Multi-Pass Graph Streaming Lower Bounds for Cycle Counting, MAX-CUT, Matching Size, and Other Problems

Sepehr Assadi∗ Gillat Kol† Raghuvansh R. Saxena‡ Huacheng Yu§

Abstract

Consider the following gap cycle counting problem in the streaming model: The edges of a 2-regular n-vertex graph G are arriving one-by-one in a stream and we are promised that G is a disjoint union of either k-cycles or 2k-cycles for some small k; the goal is to distinguish between these two cases using a limited memory. Verbin and Yu [SODA 2011] introduced this problem and showed that any single-pass streaming algorithm solving it requires \( n^{1-\Omega(1/k)} \) space. This result and the proof technique behind it—the Boolean Hidden Hypermatching communication problem—has since been used extensively for proving streaming lower bounds for various problems, including approximating MAX-CUT, matching size, property testing, matrix rank and Schatten norms, streaming unique games and CSPs, and many others.

Despite its significance and broad range of applications, the lower bound technique of Verbin and Yu comes with a key weakness that is also inherited by all subsequent results: the Boolean Hidden Hypermatching problem is hard only if there is exactly one round of communication and, in fact, can be solved with logarithmic communication in two rounds. Therefore, all streaming lower bounds derived from this problem only hold for single-pass algorithms. Our goal in this paper is to remedy this state-of-affairs.

We prove the first multi-pass lower bound for the gap cycle counting problem: Any p-pass streaming algorithm that can distinguish between disjoint union of k-cycles vs 2k-cycles—or even k-cycles vs one Hamiltonian cycle—requires \( n^{1-1/k\Omega(1/p)} \) space. This makes progress on multiple open questions in this line of research dating back to the work of Verbin and Yu.

As a corollary of this result and by simple (or even no) modification of prior reductions, we can extend many of previous lower bounds to multi-pass algorithms. For instance, we can now prove that any streaming algorithm that \((1 + \varepsilon)\)-approximates the value of MAX-CUT, maximum matching size, or rank of an n-by-n matrix, requires either \( n^{\Omega(1)} \) space or \( \Omega(\log (1/\varepsilon)) \) passes. For all these problems, prior work left open the possibility of even an \( O(\log n) \) space algorithm in only two passes.
Contents

1 Introduction 1
   1.1 Background and Motivation Behind Our Work ................. 1
   1.2 Our Contributions ....................................... 2
   1.3 Our Techniques ........................................... 4
   1.4 Erratum .................................................. 4

2 Technical Overview 5
   2.1 The Pointer Chasing Aspect of OMC ......................... 5
   2.2 The One Round Lower Bound ............................... 6
   2.3 The Round Elimination Argument ........................... 8

3 Preliminaries and Problem Definition 9
   3.1 Notation .................................................. 9
   3.2 Layered Graphs and Nested Block Graphs .................... 10
   3.3 The Final-Matching Testing Problem ........................ 11
   3.4 Main Result .............................................. 11
   3.5 The Proof of Result 1 .................................... 13

4 Lower Bound for One Round (Lemma 3.4) 15
   4.1 Block Matching of a Single Vertex After the First Round (Lemma 4.1) .... 16
   4.2 Proof of Lemma 4.4 ....................................... 21

5 Lower Bound For Multiple Rounds 24

6 Graph Streaming Lower Bounds and Beyond 29
   6.1 MAX-CUT .................................................. 30
   6.2 Maximum Matching Size .................................... 31
   6.3 Maximum Acyclic Subgraph .................................. 32
   6.4 Minimum Spanning Tree .................................... 32
   6.5 Property Testing of Connectivity, Bipartiteness, and Cycle-freeness .......... 33
   6.6 Matrix Rank and Schatten Norms ............................ 34
   6.7 Sorting-by-Block-Interchange ................................ 35

A Tools from Information Theory 43
   A.1 Auxiliary Information Theory Lemmas ....................... 45

B Further Background on Boolean Hidden Hypermaching 49
1 Introduction

Graph streaming algorithms process graphs as a sequence of edges under the usual constraints of the streaming model, i.e., by making one or a few passes over the input and using a limited memory. There are two main areas of research on graph streams: (i) the semi-streaming algorithms that use $O(n \cdot \text{polylog}(n))$ space for $n$-vertex graphs and target problems on (dense) graphs such as finding MST [42, 73], large matchings [3, 6, 10, 45, 49, 60, 72], spanners and shortest paths [15, 16, 28, 39, 40, 43, 52], sparsifiers and minimum cuts [2, 4, 65, 66, 78], maximal independent sets [8, 33, 48], graph coloring [8, 19], and the like; and (ii) the $o(n)$-space streaming algorithms that use polylog($n$) space and aim to estimate properties of (sparse) graphs such as max-cut value [21, 62–64, 67], maximum matching size [9, 31, 35, 41, 61, 74, 75], number of connected components [55], subgraph counting [14, 18, 24, 25, 34, 58, 76], property testing [37, 55, 77, 81], and others (this is by no means a comprehensive list). We will solely focus on latter algorithms in this paper.\footnote{We shall remark that the challenges for these two classes of algorithms are somewhat different as the former ones aim to compress the “edge-space” of the graph, while the latter ones focus on the “vertex-space” (see, e.g. [52]).}

We study the following problem (or rather family of problems) in the graph streaming model: Given a 2-regular graph $G = (V,E)$, decide whether $G$ consists of “many short” cycles or all cycles of $G$ are “rather long”. This problem can be seen as a robust version of cycle counting problems (similar-in-spirit to property testing, see, e.g. [17, 30]). More importantly, this problem turns out to be an excellent intermediate problem for studying the limitations of streaming algorithms.

1.1 Background and Motivation Behind Our Work

The cycle counting problem we study in this paper was first identified by Verbin and Yu [85] for proving streaming lower bounds for string problems. In the gap cycle counting problem of [85], we are given a graph $G = (V,E)$ and a parameter $k$ and are asked to determine if $G$ is a disjoint union of $k$-cycles or $2k$-cycles. Verbin and Yu proved that any single-pass streaming algorithm for this problem requires $n^{1-O(1/k)}$ space and used this to establish lower bounds for several other problems.

As most other streaming lower bounds, the proof of [85] is through communication complexity. The authors first introduced the following Boolean Hidden Hypermatching (BHH) problem: In BHH$_{n,t}$, Alice gets a vector $x \in \{0,1\}^n$ and Bob gets a perfect $t$-hypermatching $M$ on the $n$ coordinates of $x$. We are promised that the $(n/t)$-dimensional vector of parity of $x$ on hyperedges of $M$, i.e., $Mx = (\oplus_{i=1}^{n/t} x_{M_{i,t}}, \oplus_{i=1}^{n/t} x_{M_{i,t+1}}, \ldots, \oplus_{i=1}^{n/t} x_{M_{n/t}})$ is either $0^{n/t}$ or $1^{n/t}$; the goal is to distinguish between these two cases using limited communication. Building on the Fourier analytic approach of [47], Verbin and Yu gave an $\Omega(n^{1-1/t})$ lower bound on the communication complexity of this problem when only Alice can communicate. The lower bound for cycle counting then follows from a rather straightforward reduction from BHH, which in turn implies the other lower bounds in [85].

The BHH problem has since been extensively used for proving space lower bounds for streaming algorithms either through direct reductions [22, 26, 41, 54, 55, 67, 70], as a building block for other variants [9, 53, 62], or as a source of inspirations and ideas [57, 63, 64]. For instance,

(i) An $n^{1-O(\varepsilon)}$ space lower bound for $(1+\varepsilon)$-approximation algorithms of MAX-CUT by Kogan and Krauthgamer [67] and Kapralov, Khanna, and Sudan [62], which culminated in the optimal lower bound of $\Omega(n)$ for better-than-2 approximation by Kapralov and Kraunch [64] (see also [63] who proved the first $\Omega(n)$ space lower bound for $(1+\Omega(1))$-approximation);

(ii) An $n^{1-O(\varepsilon)}$ space lower bound for $(1+\varepsilon)$-approximation of maximum matching size by Esfandiari et al. [41] and Bury and Schwiegelshohn [26] which was extended to Schatten norms of $n$-by-$n$ matrices by Li and Woodruff [70] and Braverman et al. [22, 23];
(iii) An $n^{1-O(\varepsilon)}$ space lower bounds for several property testing problems such as connectivity, cycle-freeness, and bipartiteness by Huang and Peng [55].

We discuss further background on the BHH problem and list several of its other implications in Appendix B. Indeed, owing to all these implications, BHH has found its way among the few canonical communication problems—alongside Index [1,68], Set Disjointness [13,59,83], and Gap Hamming Distance [29,56]—for proving streaming lower bounds.

Yet, despite its significance and wide range of applications, BHH comes with a major weakness: BHH is a highly asymmetric problem and thus its lower bound is inherently one-way; Bob can simply send any of his hyperedges in an $O(t \cdot \log n)$-bit message which allows Alice to solve the problem. Consequently, all aforementioned lower bounds obtained from BHH in this line of work (and its many variants and generalizations) only hold for single-pass algorithms. As a result, we effectively have no knowledge of limitations of multi-pass streaming algorithms for these problems, despite the significant attention given to multi-pass algorithms lately (see, e.g. [18,21–23,34,41,76]). This raises the following fundamental question:

Can we prove a multi-round analogue of the Boolean Hidden Hypermatching lower bound that allows for obtaining multi-pass graph streaming lower bounds?

Indeed, this question and its closely related variants have already been raised several time in the literature [20,44,53,67,85] starting from the work of Verbin and Yu.

1.2 Our Contributions

Our main contribution is a multi-round lower bound for the gap cycle counting problem, in fact, in an “algorithmically simpler” form, which we call the One-or-Many Cycles (OMC) problem. We then show that by using this problem and simple (or even no) modification of prior reductions, we can extend many of previous lower bounds to multi-pass algorithms.

The One-or-Many Cycles Communication Problem

Problem 1 (One-or-Many Cycles (OMC)). Let $n,k \geq 1$ be even integers where $n$ divides $k$. In $\text{OMC}_{n,k}$, we have a 2-regular bipartite graph $G = (L,R,E)$ on $n$ vertices. Edges of $G$ consist of two disjoint perfect matchings $M_A$ and $M_B$, given to Alice and Bob, respectively. We are promised that either (i) $G$ consists of a single Hamiltonian cycle (Yes case) or (ii) $G$ is a collection of $(n/k)$ vertex-disjoint cycles of length $k$ (No case). The goal is to distinguish between these two cases.

OMC can be seen as the most extreme variant of cycle counting problems: in the No case, the graph consists of many short cycles, while in the Yes case, the entire graph is one long Hamiltonian cycle. This, at least intuitively, makes this problem “easiest” algorithmically (most suitable for reductions) and “hardest” for proving lower bounds (our lower bounds extend immediately to many other cycle counting problems including the $k$-vs-$2k$-cycle problem; see Remark 3.8).

The following is our main result in this paper.

**Result 1.** For any even integer $k > 0$ and integer $r = o(\log k)$, any communication protocol (deterministic or randomized) for $\text{OMC}_{n,k}$ in which Alice and Bob send up to $r$ messages, i.e., an $r$-round protocol, requires $n^{1-O(k^{-1/r})}$ communication.

Our lower bound in Result 1 demonstrates a tradeoff between the communication complexity of the OMC problem and the allowed number of rounds. In particular, an immediate corollary of Result 1 is that either $\Omega(\log k)$ rounds or $n^{\Omega(1)}$ communication is needed for solving $\text{OMC}_{n,k}$. 

2
Let us now briefly compare our Result 1 with [85]. By setting \( r = 1 \), we already recover the result of [85] on \( n^{1 - O(1/k)} \) communication lower bound for cycle counting problem (up to the hidden-constant in the \( O \)-notation in the exponent), but this time for the algorithmically easier problem of distinguishing \( k \)-cycles from a Hamiltonian cycle. Prior to our work, no lower bounds were known for this problem even for one-round protocols and in fact this was left as an open problem in [85, Conjecture 5.1]. But much more importantly, Result 1 now gives a \textit{multi-round} lower bound for OMC (and other problems such as \( k \)-vs-2\( k \)-cycle), making progress on another open problem of [85, Conjecture 5.4]. We note that our tradeoff does not match the conjecture in [85] and it remains a fascinating open question to determine the “right” tradeoff for this problem.

### Streaming Lower Bounds from OMC

The OMC problem is able to capture the essence of many of previous streaming lower bounds proven via reductions from the BHH problem. In fact, as we shall see, these reductions often become even easier now that we are working with the OMC problem considering it has a more natural graph-theoretic definition compared to BHH. But more importantly, we can now use the lower bound for OMC in Result 1 to give \textit{multi-pass} streaming lower bounds.

Before listing our results, let us give a concrete example of a lower bound that we can now prove using the OMC problem to emphasize the simplicity of the reductions.

**Example: Property testing of connectivity.** Huang and Peng [55] gave a reduction from BHH to prove that any single-pass streaming algorithm for property testing of connectivity, namely, deciding whether a graph is connected or it requires at least \( \varepsilon \cdot n \) more edges to become connected, needs \( n^{1 - O(\varepsilon)} \) space. We use a reduction from OMC to extend this lower bound to multiple passes.

Let \( k := \frac{1}{2\varepsilon} \) and \( G \) be a graph in the \( \text{OMC}_{n,k} \) problem. In the Yes case, \( G \) is a Hamiltonian cycle and is thus already connected. On the other hand, in the No case, \( G \) consists of \( n/k \) disjoint cycles and thus to be connected requires \( n/k - 1 > \varepsilon n \) edges. We can run any streaming algorithm on input graphs of \( \text{OMC}_{n,k} \) by Alice and Bob creating the stream \( M_A \) appended by \( M_B \) and passing along the memory content of the algorithm to each other. As such, using an algorithm for connectivity, the players can also solve \( \text{OMC}_{n,k} \). By Result 1, this implies that any \( p \)-pass streaming algorithm for connectivity requires \( n^{1 - \Theta(1/p)} \) space. Interestingly, not only this reduction gives us a multi-pass lower bound, but also it is arguably simpler that the reduction from BHH.

We prove the following lower bounds by reductions from OMC and our lower bound in Result 1.

**Result 2.** Let \( g(\varepsilon, p) := O(\varepsilon^{c/p}) \) for some large enough absolute constant \( c > 0 \). Then, any \( p \)-pass streaming algorithm for any of the following problems on \( n \)-vertex graphs requires \( n^{1 - g(\varepsilon, p)} \) space (the references below list the previous single-pass lower bounds for the corresponding problem):

(i) \((1 + \varepsilon)\)-approximation of MAX-CUT in sparse graphs (cf. [62–64, 67]);

(ii) \((1 + \varepsilon)\)-approximation of maximum matching size in planar graphs (cf. [26, 41]);

(iii) \((1 + \varepsilon)\)-approximation of the maximum acyclic subgraph in directed graphs (cf. [54]);

(iv) \((1 + \varepsilon)\)-approximation of the weight of a minimum spanning tree (cf. [42, 55]);

(v) property testing of connectivity, bipartiteness, and cycle-freeness for parameter \( \varepsilon \) (cf. [55]).

Our lower bounds can be extended beyond graph streams. For instance, we can also prove lower bounds of \( n^{1 - g(\varepsilon, p)} \) space for \((1 + \varepsilon)\)-approximation of rank and other Schatten norms of \( n \)-by-\( n \) matrices (cf. [22, 26, 70]), or sorting-by-block-interchange on \( n \)-length strings (cf. [85]).
A simple corollary of these lower bounds is that for any of these problems, any streaming algorithm requires either \( \Omega(\log (1/\epsilon)) \) passes or \( n^{\Omega(1)} \) space. Prior to our work, even an \( O(\log n) \) space algorithm in two passes was not ruled out for any of these problems. These results settle or make progress on multiple open questions in the literature regarding the multi-pass streaming complexity of gap cycle counting [85], MAX-CUT [20,67], and streaming CSPs [53]. We postpone the exact details of our results and further backgrounds in this part to Section 6.

To conclude, we believe that Result 1 and Result 2 identify OMC as a natural multi-round analogue of the BHH problem (as was asked in prior work [67, 85]), answering our motivating question. This is indeed the main conceptual contribution of our work.

### 1.3 Our Techniques

We briefly mention the techniques used in our paper here and postpone further details to the technical overview of our proof in Section 2.

The lower bound for BHH in [85] and other variants in this line of research [53,57,62–64] all relied heavily on techniques from Fourier analysis on the Boolean hypercube. In contrast, our proofs in this paper solely relies on tools from information theory.

The first main technical ingredient of our work is a novel round-elimination argument for OMC. Typical round-elimination arguments for similar problems such as pointer chasing on graphs [27, 38,79,80,82,87] “track” the information revealed about a particular path inside the graph, ensuring that the player who is speaking next is unaware of which pointer to chase. On the other hand, we crucially need to track the information revealed about multiple vertices at the same time (to account for the strong promise in the input instance). As such, our proof takes a different approach. We first show that after the first message of the protocol, there is a “large” minor of the graph—obtained by contracting “short” paths—that still “looks random” to players. Eliminating a communication round at this point then boils down to embedding a hard instance for \( r \)-round protocols inside this minor of the hard instance of \( (r+1) \)-round protocols; this in particular involves embedding a “lower dimensional” instance on smaller number of vertices inside a “higher dimensional” one (as a graph minor and not a subgraph).

Our second main technical ingredient is the proof of existence of this “random looking” graph minor after the first message of the protocol. At the heart of this part is an argument showing that after a single message, the joint distribution of all the vertices reachable from a fixed set of \( \approx (n/k) \) vertices by taking a constant number of edges remains almost identical to its original distribution. This is done by first proving a one-round “low-advantage” version of standard pointer chasing lower bounds: For each starting vertex \( v \) in the graph, after one message, the distribution of the unique vertex which is at distance \( c \) of \( v \) is \( \approx n^{-\Omega(c)} \)-close to its original (uniform) distribution. The proof is based on bounding the \( \ell_2 \) norm of the distribution of this unique vertex, and applying a direct product type argument for \( \Theta(c) \) different sub-problems, each corresponding to going “one more edge away” from the starting vertex. The final proof for the joint distribution of the targets of \( \approx (n/k) \) vertices is done through a series of reductions from this single-vertex variant.

### 1.4 Erratum

The previous version of this paper had a mistake in one of the technical lemmas: all the main results of the paper are correct as stated, but Lemma 4.2 of Section 5 of the previous version has a subtle flaw (which is in fact not in the main part of the proof, but rather a false observation made in the final part to generalize the proof). Lemma 4.2 was not a significant result in and of itself, but was used as a building block for our main result. In this version, we present a modified
proof of our results. Most of the building blocks and the overall structure of the proofs are the same as before, except we no longer rely on the false observation in Lemma 4.2 and instead prove a somewhat weaker variant of this lemma; the remaining changes are there to allow for working with this weaker statement in the rest of the proof. See Remark 3.6 for some more context.

2 Technical Overview

We present a streamlined overview of our technical approach for proving Result 1 in this section (we leave the details of our reductions in Result 2 to Section 6). We emphasize that this section oversimplifies many details and the discussions will be informal for the sake of intuition.

Before getting to the discussion of our lower bound, it helps to consider what are some natural ways for Alice and Bob to solve OMC\(_{n,k}\). At one extreme, there is a “sampling” approach: Alice can randomly sample \(O(n^{1-1/k})\) edges from her input and send them to Bob; the (strong) promise of the problem ensures that in the No-case, Bob will, with constant probability, see an entire \(k\)-cycle in the graph and thus can distinguish this from the Yes-case. At the other extreme, there is a “pointer chasing” approach: the players can start from any vertex of the graph and simply “chase” a single (potential) \(k\)-cycle one edge per round and in (roughly) \(k\) rounds solve the problem. And then there are different interpolations between these two, for instance by chasing \(O(n^{1-1/\sqrt{k}})\) random vertices in \(\sqrt{k}\)-rounds. Our lower bound has to address all these approaches simultaneously.

2.1 The Pointer Chasing Aspect of OMC

Let us start with the “pointer chasing” aspect of our lower bound. Suppose we put the following additional structure on the input to Alice and Bob:

(i) The input graph \(G\) consists of \(k\) layers \(V_1, \ldots, V_k\) of size \(m = n/k\) with \(k\) perfect matchings \(M_1, \ldots, M_k\) between them, where \(M_i\) is between \(V_i\) and \(V_{(i+1 \mod k)}\).

(ii) For any \(v \in V_1\), define \(P(v) \in V_k\) as the unique vertex reachable from \(v\) in \(G\) using only \(M_1, \ldots, M_{k-1}\). We promise that either: (a) for all \(v_i \in V_1\), \(P(v_i)\) connects back to \(v_i\) in the matching \(M_k\), i.e., \(M_k \circ P = I\) where \(I\) is the identity matching; or (b) for all \(v_i \in V_1\), \(P(v_i)\) connects to \(v_{(i+1 \mod m)}\) in \(M_k\), i.e., \(M_k \circ P = S_{+1}\), where \(S_{+1}\) is the cyclic-shift-by-one matching.

(iii) The input to the players are then alternating matchings from this graph, namely, Alice receives even-indexed matchings \(M_2, M_4, M_6, \ldots\), and Bob receives odd-indexed ones \(M_1, M_3, M_5, \ldots\).

It is easy to verify that for even values of \(k\), these graphs form valid inputs to OMC\(_{n,k}\) problem (with case (a) corresponding to the No-case and (b) corresponding to the Hamiltonian cycle case).

Under this setting, we can interpret OMC\(_{n,k}\) as some pointer chasing problem: for some vertex \(v \in V_1\), the players need to “chase the pointers”

\[
M_1(v), \quad M_2 \circ M_1(v), \quad \ldots, \quad M_{k-1} \circ M_{k-2} \circ \cdots \circ M_1(v)
\]

to reach \(P(v) \in V_k\); then check whether \(P(v)\) connects back to \(v\) in \(M_k\) or not.

There are however several differences between a typical pointer chasing problem (see, e.g. [7,27, 50,52,79,87] for many different variants) and our problem. Most important among these is that the strong promise in the input effectively means there is no single particular pointer that the players need to chase—all they need to do is to figure out \(P(v)\) for some \(v \in V\) after communicating the messages (this is on top of apparent issues such as players being able to chase pointers from “both ends” and the like). We elaborate more on this below.
For intuition, let us consider the following specialized protocol \( \pi \): the players first completely ignore the matching \( M_k \) and instead aim to “learn” the mapping \( P = M_{k-1} \circ M_{k-2} \circ \cdots \circ M_1 \); only then, they will look at \( M_k \) and check whether \( P \circ M_k \) is \( I \) or \( S_+ \), corresponding to cases (a) or (b) above. Under this view, we can think of the following two-phase problem:

1. Given \( M_1, \ldots, M_{k-1} \) sampled independently and uniformly at random, the players run \( \pi \) with transcript \( \Pi \) which induces a distribution \( \text{dist}(P | \Pi) \) for \( P = M_{k-1} \circ \cdots \circ M_1 \);

2. Only then, given a matching \( M_k \) sampled uniformly from just two choices \( \{I \circ P^{-1}, S_+ \circ P^{-1}\} \) implied by cases (a) or (b), they need to determine which case \( M_k \) belongs to.

For such a protocol to fail to solve the problem better than random guessing, we should have that:

\[
\mathbb{E}_{\Pi} \|\text{dist}(I \circ P^{-1} | \Pi) - \text{dist}(S_+ \circ P^{-1} | \Pi)\|_{\text{tvd}} = o(1),
\]

(here and throughout, \( \Pi \) is the transcript of the protocol)

as the players are getting one sample, namely, \( M_k \), from one of the two distributions and thus should not be able to distinguish them with one sample\(^2\). As such, the task of proving a lower bound essentially boils down to showing that for a “small” round and communication protocol \( \pi \),

\[
\mathbb{E}_{\Pi} \|\text{dist}(P | \Pi) - \text{dist}(P)\|_{\text{tvd}} = o(1),\tag{1}
\]

namely, that the protocol cannot change the distribution of the entire mapping \( P \) by much; this should be contrasted with typical lower bounds for pointer chasing that require that distribution of a single pointer \( P(v) \) for \( v \in V_1 \) does not differ considerably after the communication.

This outline oversimplifies many details. Most importantly, it is not at all the case that the only protocols that solve the problem adhere to the special two-phase approach mentioned above. Indeed, the input of players are highly correlated in the problem and this can reveal information to the players. Consequently, in the actual lower bound, we need to handle these correlation throughout the entire proof. In particular, we need stronger variant of (1) that shows the value of distribution \( P | \Pi \) on two particular points (rather than two marginally random points) in the support are close. We postpone these details to the actual proof and for now focus on the lower bound for (1) which captures the crux of the argument.

### 2.2 The One Round Lower Bound

We first prove a stronger variant of (1) for one round protocols. Suppose \( M_1, \ldots, M_{k-1} \) are sampled uniformly at random and \( M_B \) denotes the input of Bob among these matchings. Then, we show that if Alice sends a single message \( \Pi \) of size \( C = o(m) \) (recall that \( m = n/k \)), we will have:

\[
\mathbb{I}(P ; M_B, \Pi) \leq m^{\Theta(1)} \cdot \left( \frac{C}{m} \right)^{\Theta(k)}.
\]

Let us interpret this bound: the mutual information between \( P \) and \( (M_B, \Pi) \) is a measure of how much the distribution of \( P \) is affected by the extra conditioning on \( M_B, \Pi \) (see Fact A.4); in particular, if we set \( C = m^{1-\Theta(1/k)} \) in the above bound, we get that the RHS is \( o(1) \) for large enough \( k \) which in turn implies that distribution of \( P \) conditioned on both \( M_B, \Pi \), is very close to its original distribution, implying (1) (in a stronger form). The proof of (2) consists of two parts.

\(^2\)Note that technically we should also condition on the input of the player outputting the final answer but for simplicity, we will ignore that in this discussion for now.
Part One: The main part of the proof is to consider a “low advantage” variant of pointer chasing:

− **Low-advantage pointer chasing:** Given a fixed vertex \( v \in V_1 \) and assuming we are only allowed one round of communication, what is the probability of guessing \( P(v) \in V_k \) correctly?

Standard pointer chasing lower bounds such as \([79, 87]\) imply that the answer for a \((< k)\)-round protocol with communication cost \( C \) is roughly \( \frac{1}{m} + \frac{C}{km} \). We on the other hand prove a much stronger bound but only for one-round protocols which is roughly \( \frac{1}{m} + \left( \frac{C}{m} \right) \Theta(k) \).

The proof of this part is one of the main technical contributions of our work. For our purpose it would be easier to bound the \( \ell_2 \)-norm of the vector \( P(v) \), i.e., show that:

\[
\mathbb{E}_{M_B, \Pi} \|P(v) \mid M_B, \Pi\|_2^2 \leq \frac{1}{m} + \left( \frac{C}{m} \right) \Theta(k),
\]

which can be used to bound the advantage of the protocol over random guessing.

It turns out the key to bounding the LHS above is to understand the “power” of message \( \Pi \) in changing the distribution of collections of edges chosen from disjoint matchings in the input of Alice. Let \( S \subseteq \{2, 4, \ldots, k-2\} \) be a set of indices of Alice’s matchings and \( v_S = (\hat{e}_{i_1}, \ldots, \hat{e}_{i_{|S|}}) \) denote a collection of “potential” edges in \( M_{i_1}, M_{i_2}, \ldots, M_{i_{|S|}} \) (i.e., pairs of vertices which may or may not be an edge in each matching). In particular, we can bound the LHS of (3) by

\[
\mathbb{E}_{\Pi} \left[ \sum_{\forall S} \Pr \left( \hat{e}_{i_1} \in M_{i_1} \land \cdots \land \hat{e}_{i_{|S|}} \in M_{i_{|S|}} \mid \Pi \right) \right] \leq \left( \frac{Cm}{|S|^2} \right)^{\Theta(|S|/2)}. \tag{5}
\]

Plugging in (5) inside (4) then prove the “low advantage” lower bound we want in (3).

The statement (and the proof of) (5) can be seen as some direct product type result: When \(|S| = 1\), we are simply bounding the (square of the) probability that a potential edge belongs to the graph of Alice conditioned on an “average” message; considering each matching \( M_i \) is a random permutation of size \([m]\) and the message reveals only \( C = o(m) \) bits about it, we expect only a small number of edges to have a “high” probability of appearing in \( M_i \) conditioned on \( \Pi \) (see Lemma A.13). Our bounds in (5) then show that repeating this task for \(|S| \) times, namely, increasing the probability of an entire \(|S|\)-tuple \( v_S \) of edges, becomes roughly \(|S| \) times less likely.

Part Two: Our lower bound in Eq (3) can also be interpreted as bounding:

\[
\mathbb{I}(P(v) ; M_B, \Pi) \leq \left( \frac{C}{m} \right) \Theta(k), \tag{6}
\]

for a fixed vertex \( v \in V_1 \), namely, a single-vertex version of the bound we want in (2). We obtain the final bound using a series of reductions from this. In particular, by chain rule:

\[
\mathbb{I}(P ; M_B, \Pi) = \sum_{v \in V_1} \mathbb{I}(P(v) ; M_B, \Pi \mid P(v_1), \ldots, P(v_{i-1})).
\]
For the first $m/2$ terms in the sum above, we can show that the bounds in (6) continue to hold even conditioned on the new $\mathcal{P}$-values; this is simply because even conditioned on these values, at least half of each matching is “untouched” and thus we can apply the previous lower bound to the underlying subgraph with ($\geq m/2$)-size layers instead. This argument however cannot be extended to all values in the sum simply because the size of layers are becoming smaller and smaller through this conditioning. To handle this in our actual proof, we modify our graph by shrinking the first and the last layers by a constant factor and generalize all our arguments to this modified graph. We postpone the details of this part to the actual proof but note here that this essentially amounts to “getting rid” of the last $m/2$ terms.

2.3 The Round Elimination Argument

The next key ingredient of our proof is a round elimination argument for “shaving off” the rounds in any $r$-round protocol one by one, until we end up with a 0-round protocol that can still solve a non-trivial problem; a contradiction.

A typical round elimination argument for pointer chasing shows that after the first message of the protocol, the distribution of next immediate pointer to chase (namely, $M_2 \circ M_1 (v)$ when chasing $M_k \circ \cdots \circ M_1 (v)$) is still almost uniform as before. Thus, the players now need to solve the same problem with one less round and one less matching. Unfortunately, such an approach does not suffice for our purpose in proving (1) in which chasing any pointer solves the problem.

Our round elimination argument thus takes a different route. We show that after the first message of the protocol, the joint distribution of compositions of matchings still look almost uniform to Bob. Notice that the main difference in (7) compared to (2) that the bounds are now applied to each block of length $c$ in the graph, not the entire $k$ layers.

Now let us see how we can use this to eliminate one round of the protocol. This is done through an embedding argument in which we embed an instance $G_{r-1}$ of the problem on $k_{r-1}$ layers inside a graph $G_r$ of $k_r$ layers conditioned on the first message, and run the protocol $\pi$ for $G_r$ from the second round onwards to obtain an $(r-1)$-round protocol $\Pi_r$. The embedding is as follows.

**Embedding $G_{r-1}$ inside $G_r$ | $\Pi_1$.** Let $M_1, \ldots, M_{k_{r-1}}$ be the inputs to Alice and Bob in $G_{r-1}$. In the protocol $\theta$, the players first use public randomness to sample a message $\Pi_1$ from the distribution induced by $\pi$ on $G_r$. We would now like to sample a graph $G_r | \Pi_1$ such that:

(i) for every $i \in [k_{r-1}]$, we have $P_i = M_i$, i.e., the composition of the $i$-th block of $c$ consecutive matchings in $G_r$ looks the same as the matching $M_i$;

(ii) Alice in $\theta$ gets the input of Bob in $\pi$ in $G_r$ and Bob in $\theta$ gets the input of Alice in $\pi$.

Assuming we can do this, Alice and Bob in $\theta$ can continue running $\pi$ on $G_r | \Pi_1$ as they both know the first message of $\pi$ and by property (ii) above have the proper inputs; moreover, by
property (ii), $M_{k_r-1} \circ \ldots \circ M_1 = P_{k_r-1} \circ \ldots \circ P_1 = P$, i.e., the distribution of pointers they would like to chase in both $G_r$ and $G_{r-1}$ is the same. Thus if $\pi$ was able to change the distribution of $P$ in $G_r$ in $r$ rounds, then $\theta$ can also change the distribution of $P$ in $G_{r-1}$ in $(r-1)$ rounds.

Of course, we cannot hope to straightaway perform the sampling above without any communication between the players as (i) $\Pi_1$ correlates the distribution of $P_1, \ldots, P_{k_{r-1}}$ in $G_r \mid \Pi_1$ (even though they were independent originally), and (ii) Alice and Bob in $\theta$ know only half of $P_1, \ldots, P_{k_{r-1}}$ each (dictated by their input in $G_{r-1}$). This is where we use (7). Intuitively, since the distribution of $P_1, \ldots, P_{k_{r-1}} \mid \Pi_1, M_B$ has not changed dramatically (and not at all if we condition on $M_A$ instead of $M_B$ since Alice is the sender of $\Pi_1$), we can design a sampling process based on a combination of public and private randomness that "simulates" sampling

$$G_r \sim G_r \mid \Pi_1, P_1 = M_1, \ldots, P_{k_{r-1}} = M_{k_{r-1}}$$

by instead sampling from the product distribution

$$G_r \sim \prod_{i \in [k_{r-1}]} \text{matchings } M_{i,1} \ldots, M_{i,c} \text{ in } i\text{-th block of } G_r \mid \Pi_1, P_1 = M_i,$$

while obtaining the same answer as $\pi$ up to a negligible factor of $1/\text{poly}(m)$ error.

To conclude, if the communication cost of the protocol is only $C = m^{1-\Theta(1/c)}$, we can shave off all the $r$-rounds of the protocol, while shrinking the number of layers in input graphs by a factor of $c$ each time; by the choice of $k_r = c^r$, we will eventually end up with a $0$-round protocol on a non-empty graph which cannot change the distribution of corresponding $P$ at all. Tracing back the argument above then implies that the original $r$-round protocol should not be able to change the distribution of its own mapping $P$ by more than $O(r/\text{poly}(m))$ as desired. Rewriting $c$ as $k^{1/r}$, we obtain an $m^{1-\Theta(1/k^{1/r})}$ lower bound on the communication cost of protocols for (1).

3 Preliminaries and Problem Definition

3.1 Notation

Sequences. Unless stated otherwise, we use the letters $S, T$ to denote sequences. The notation $S \| T$ will denote the concatenation of the sequences $S$ and $T$ and $|S|$ denotes the length of $S$. For a sequence $S$ such that $|S| > 0$, we use $S[\cdot:-1]$ to denote the sequence $S$ without its last element. For $i \in [|S|]$, we use $S_{\leq i}$ to denote the sequence with the first $i$ elements of $S$. Define $S_{> i}$ analogously. Finally, for some $s$ and $c > 0$, the notation $s^{(c)}$ denotes the $c$-length sequence with all elements equal to $s$.

Random variables. When there is room for confusion, we use sans-serif letters for random variables (e.g. $A$) and normal letters for their realizations (e.g. $A$). For a random variable $A$, we use $\text{supp}(A)$ to denote the support of $A$ and $\text{dist}(A)$ to denote the distribution of $A$. When it is clear from the context, we may abuse the notation and use $A$ and $\text{dist}(A)$ interchangeably. By norm $||A||$ of a random variable with distribution $\mu$ and support $a_1, \ldots, a_m$, we mean the norm of the vector $(\mu(a_1), \ldots, \mu(a_m))$.

Information theory. For random variables $A, B$, we use $\mathbb{H}(A)$ to denote the Shannon entropy of $A$, $\mathbb{H}(A \mid B)$ to denote the conditional entropy, and $\mathbb{I}(A ; B)$ to denote the mutual information. Similarly, for two distributions $\mu$ and $\nu$ on the same support, $||\mu - \nu||_{\text{tvd}}$ denotes their total variation distance and $\mathbb{D}(\mu \mid \mid \nu)$ is their KL divergence. Appendix A contains the definitions and standard properties of these notions as well as some auxiliary lemmas that we prove in this paper.
3.2 Layered Graphs and Nested Block Graphs

We start by defining a layered graph.

**Definition 3.1.** Let \( k > 0 \) and \( T = \{ t_i \}_{i=0}^{k} \) be a sequence of positive integers. A \( T \)-layered graph \( G = (V, E) \) is such that \( |V| = \sum_{i=0}^{k} t_i \) and the following holds:

If \( V = V_0 \uplus V_1 \uplus \cdots \uplus V_k \) is the lexicographically smallest partition of \( V \) into sets of sizes \( t_0, t_1, \cdots, t_k \) respectively, then the set \( E \) can be partitioned as \( E = E_1 \uplus \cdots \uplus E_k \) where for all \( i \in [k] \), \( E_i \) is a matching of size \( \min(t_{i-1}, t_i) \) between \( V_{i-1} \) and \( V_i \).

Observe that the partition \( E = E_1 \uplus \cdots \uplus E_k \) is unique. When we wish to emphasize the partitions, we write \( G = (\bigcup_{i=0}^{k} V_i, \bigcup_{i=1}^{k} E_i) \). We call the sets \( V_i \) the ‘layers’ of \( G \), with \( V_0 \) being the first and \( V_k \) being the last layer. Often when we talk about layered graphs, we will have a fixed set of layers in mind and consider different sets of edges between these layers. In particular, when we talk about a random layered graph, we mean a graph chosen uniformly at random amongst all layered graphs with those layers. We use \( \mathcal{L}_T \) to denote the uniform distribution over all \( T \)-layered graphs and omit the subscript when it is clear from context.

As the edges in a layered graph form a matching between adjacent layers, any vertex in any layer is connected to at most one vertex in any other layer. For \( i < i' \in \{0\} \cup [k] \), we define the function \( G_{i \rightarrow i'} : V_i \rightarrow V_{i'} \) to be such that for all \( v \in V_i \), \( G_{i \rightarrow i'}(v) \) is either the unique vertex in layer \( V_{i'} \) that \( v \) connects to, or is \( \perp \), if no such vertex exists. We simply say \( G_{i'} \) if \( i' = i + 1 \). We say that a layered graph is **nice** if for all \( i < i' \in \{0\} \cup [k] \), the number of vertices in \( V_i \) such that \( G_{i \rightarrow i'}(v) \neq \perp \) is the largest possible. Formally,

**Definition 3.2.** Let \( k > 0 \) and \( T = \{ t_i \}_{i=0}^{k} \) be a sequence of positive integers. A \( T \)-layered graph \( G = (\bigcup_{i=0}^{k} V_i, \bigcup_{i=1}^{k} E_i) \) is said to be nice if, for all \( i < i' \in \{0\} \cup [k] \), we have

\[
|\{v \in V_i \mid G_{i \rightarrow i'}(v) \neq \perp\}| = \min_{i \leq i'' \leq i'} t_{i''}.
\]

Next we define a subset of layered graphs that we use throughout this work. Let \( c, s, k > 0 \) and \( T = \{ t_i \}_{i=0}^{k} \) be a sequence of positive integers. Define the sequence \( \text{plug}(T, c, s) \) as:

\[
\text{plug}(T, c, s) = t_0 || s(c) || t_1 || \cdots || s(c) || t_k.
\]

Namely, \( \text{plug}(T, c, s) \) is the sequence obtained by plugging \( c \) occurrences of \( s \) between any two elements of \( T \). Let \( S = \{ s_i \}_{i=1}^{r} \) be an increasing\(^3\) sequence of positive integers. Define:

\[
\text{nest}(c, S) = \begin{cases} s_1^{(2)}, & \text{if } |S| = 1, \\ \text{plug}(\text{nest}(c, S[: -1]), 2c - 1, s_{|S|}), & \text{if } |S| > 1. \end{cases}
\]

We define \( \ell_{c,S} = |\text{nest}(c, S)| - 1 \) and note that \( \ell_{c,S} = (2c)^{|S|-1} \). We drop the subscripts \( c, S \) when they are clear from context.

**Definition 3.3 (Nested block graphs).** Let \( c > 0 \) and \( S \) be an increasing sequence of positive integers. A \((c, S)\)-nested block graph is a \( \text{nest}(c, S) \)-layered graph that is nice.

Analogous to layered graphs, we shall consider random nested block graphs over the same set of vertices. We reserve \( \mathcal{D}_{c,S} \) to denote the uniform distribution over \((c, S)\)-nested block graphs and omit the subscripts when they are clear from context. For a permutation \( Z \) on \( [s_1] \) we use \( \mathcal{D}_Z^{c,S} \) to denote the distribution \( \mathcal{D}_{c,S} \) over nested block graphs \( G \) conditioned on \( G_{0 \rightarrow \ell} = Z \).

\(^3\)We adopt the convention that the sequence \( S \) will be indexed from 1 and increasing, while \( T \) will be indexed from 0 and may or may not be increasing.
Almost nested block graphs. Let $c, k > 0$ and $S = \{s_i\}_{i=1}^k$ be an increasing sequence of positive integers. Let $0 < s \leq s_1$ be given. We define the sequence $\text{nest}_l(s, c, S)$ to equal to $\text{nest}(c, S)$ except that the first element is $s$ instead of $s_1$. Similarly, the sequence $\text{nest}_r(s, c, S)$ to equal to $\text{nest}(c, S)$ except that the last element is $s$. We omit the subscript $l, r$ when it is clear from context. We define an $(s, c, S)$-almost nested block graph to be a $\text{nest}(s, c, S)$-layered graph that is nice. In other words, an $(s, c, S)$-almost nested block graph is the same as a $(c, S)$-nested block graph except that either the first or the last layer is of size $s$. We define the notation $\mathcal{D}_{s,c,S}$ and $\mathcal{D}_{s,c,S}^Z$ for almost nested block graphs analogous to our notation for nested block graphs. We omit left and right in this notation as it shall be clear from context.

3.3 The Final-Matching Testing Problem

We work in the standard two-party communication model [86] (see [69] for an overview of the standard definitions). For a sequence $T$ and $r > 0$, an $r$-round $T$-protocol $\pi$ is defined over a $T$-layered graph $G = (\bigsqcup_{i=0}^{T-1} V_i, \bigsqcup_{i=1}^{T-1} E_i)$. Alice's input is the edges $E_i$ for even $i$ and Bob's input is the edges $E_i$ for odd $i$. Both players know the partition $\bigsqcup_{i=0}^{T} V_i$. Using these inputs, Alice and Bob communicate for $r$ rounds where Alice sends a message to Bob in odd rounds and Bob sends a message to Alice in even rounds. If $r$ is even (resp. odd), then, after $r$ rounds, Alice (resp. Bob) computes an output based on its input and the transcript. We use $\|\pi\|$ to denote the communication cost of the protocol $\pi$ defined as the worst-case number of bits communicated (not including the output) between Alice and Bob in $\pi$ over any input. We also use $\pi(G)$ or $\pi(\bigsqcup_{i=1}^{T} E_i)$ to denote the output of the protocol $\pi$ when the inputs are $\bigsqcup_{i=1}^{T} E_i$. If $T = \text{nest}(c, S)$ for some sequence $S$ and $c > 0$, we may say that $\pi$ is an $r$-round $(c, S)$-protocol. Analogously, define an $r$-round $(s, c, S)$-protocol with almost nested block graphs instead of nested block graphs.

Based on these, we define the following communication problem.

Problem 2 ((Distributional) Final-Matching Testing). The FMT problem is a two-player communication problem parameterized by integers $m, c, r \geq 1$ and two fixed permutations $Y \neq N$ on $[m]$. Let $S$ be the sequence $S = m, m + \left\lfloor \frac{m}{r} \right\rfloor, m + \left\lfloor \frac{2m}{r} \right\rfloor, \cdots, 2m$.

We sample a $(c, S)$-nested block graph $G$ from the distribution $\frac{1}{2} \cdot \mathcal{D}_{c,S}^Y + \frac{1}{2} \cdot \mathcal{D}_{c,S}^N$, and the goal is to determine which is the case, i.e., whether the final matching $G_{0\rightarrow \ell}$ is $Y$ or $N$.

3.4 Main Result

Our main technical result is the following lower bound for the FMT problem. Result 1 then follows easily from this theorem as we show in Section 3.5.

Theorem 1. Let $r, m, c > 0$ be such that $m$ is odd and sufficiently large. Define the sequence $S = m, m + \left\lfloor \frac{m}{r} \right\rfloor, m + \left\lfloor \frac{2m}{r} \right\rfloor, \cdots, 2m$. For all permutations $Y, N$ on $[m]$ and all randomized $r$-round $(c, S)$-protocols $\Pi$ such that $\|\Pi\| \leq 2^{-O(c)} \cdot m^{1 - \frac{200}{c}}$, it holds that:

$$\|\text{dist}_{G \sim \mathcal{D}_{c,S}^Y} (\Pi(G)) - \text{dist}_{G \sim \mathcal{D}_{c,S}^N} (\Pi(G))\|_{\text{tvd}} \leq \frac{1}{m^{\delta}}.$$ 

The proof of Theorem 1 is by induction on $r$. The base case ($r = 1$) is captured in Lemma 3.4 while the induction step is captured in Lemma 3.5. Observe that Lemma 3.4 and Lemma 3.5 talk about almost nested block graphs while Theorem 1 talks about nested block graphs. This generalization is needed to make the induction work.
Lemma 3.4. Let $S = \{s_i\}_{i=1}^{2}$ be an increasing sequence and $c, C > 0$ be integers. For all $s \leq s_1$, injections $Y, N : [s] \to [s_1]$, and all randomized 1-round $(s, c, S)$-protocols $\Pi$ such that $||\Pi|| \leq C$, we have

$$\|\text{dist}_{G \sim D_{s,c,S}}(\Pi(G)) - \text{dist}_{G \sim D_{s,c,S}}(\Pi(G))\|_{\text{tvd}} \leq 2s^2 \cdot \left( \frac{40C}{s_2 - s} \right)^{c/6} := \Phi(s, c, s_2, C).$$

Lemma 3.5. Fix an increasing sequence $S$ and $c, C, c_0 > 0$. Assume that $|S| \geq 2$ and for all $s \leq s_2$, injections $Y', N' : [s] \to [s_2]$, and all randomized $(|S| - 2)$-round $(s, c, S_{\geq 2})$-protocols $\Pi'$ such that $||\Pi'|| \leq C$, we have:

$$\|\text{dist}_{G \sim D_{s,c,S_{\geq 2} \geq 2}}(\Pi'(G')) - \text{dist}_{G \sim D_{s,c,S_{\geq 2} \geq 2}}(\Pi'(G'))\|_{\text{tvd}} \leq \epsilon_0.$$ 

Then, for all $s \leq s_1$, injections $Y, N : [s] \to [s_1]$, and all randomized $(|S| - 1)$-round $(s, c, S)$-protocols $\Pi$ such that $||\Pi|| \leq C$, we have:

$$\|\text{dist}_{G \sim D_{s,c,S}}(\Pi(G)) - \text{dist}_{G \sim D_{s,c,S}}(\Pi(G))\|_{\text{tvd}} \leq 2c\epsilon_0 + \Phi(s, c, s_2, C),$$

where $\Phi(\cdot)$ is as defined in Lemma 3.4.

We now prove Theorem 1.

Proof of Theorem 1 (assuming Lemmas 3.4 and 3.5). Let $C = ||\Pi||$ and observe that $|S| = r + 1$. Using Lemma 3.4 once and Lemma 3.5 $r - 1$ times, we get that:

$$\|\text{dist}_{G \sim D_{c,S}}(\Pi(G)) - \text{dist}_{G \sim D_{c,S}}(\Pi(G))\|_{\text{tvd}} \leq (2c)^{r-1} \cdot \sum_{i=2}^{r} \Phi(s_{i-1}, c, s_i, C).$$

Plugging in the definition of $\Phi(\cdot)$ from Lemma 3.4, we get:

$$\|\text{dist}_{G \sim D_{c,S}}(\Pi(G)) - \text{dist}_{G \sim D_{c,S}}(\Pi(G))\|_{\text{tvd}} \leq (2c)^{r-1} \cdot 2s^2 \cdot \left( \frac{40C}{s_1 - s_{i-1}} \right)^{c/6}.$$ 

We simplify using the definition of $S$:

$$\|\text{dist}_{G \sim D_{c,S}}(\Pi(G)) - \text{dist}_{G \sim D_{c,S}}(\Pi(G))\|_{\text{tvd}} \leq (2c)^{r-1} \cdot 8s^2 \cdot \left( \frac{80rC}{m} \right)^{c/6}.$$ 

Using $C \leq 2^{-O(r)} \cdot m^{1 - \frac{200}{c}}$, we get:

$$\|\text{dist}_{G \sim D_{c,S}}(\Pi(G)) - \text{dist}_{G \sim D_{c,S}}(\Pi(G))\|_{\text{tvd}} \leq (2c)^{r-1} \cdot 8s^2 \cdot 2^{-O(r)} \cdot m^{-30} \leq \frac{1}{m^3}.$$ 

Remark 3.6. In an earlier version of this paper, we defined a similar FMT problem with the difference that the size of every layer was equal and the final matching corresponded to a perfect matching between the first and last layers. And we mistakenly claimed an analogue of Theorem 1 for this problem.

However, one can verify that in that case, if the parity of permutations $Y$ and $N$ are different, then there is a simple $O(1)$ communication protocol that can solve the problem in one-round by Alice sending the parity of her combined matchings. As such, Theorem 1 in such generality cannot hold for that version of FMT problem, hence necessitating the approach taken in the current version.
3.5 The Proof of Result 1

In this section, we formalize and prove Result 1 assuming Theorem 1.

**Theorem 2.** For even \( k > 0 \) and \( n, r > 0 \) such that \( n \) is an odd multiple of \( k \), any \( r \)-round communication protocol \( \pi \) that solves \( \text{OMC}_{n,k} \) with probability at least \( (1/2 + (k/n)^2) \) has communication cost \( \|\pi\| = 2^{-O(r)} \cdot (n/k)^{1-10^{-k-1}/r} \).

**Proof.** Let us first assume that \( k \) is a multiple of 4 and define \( m = n/k \). As \( n \) is an odd multiple of \( k \), we have that \( m \) is odd. Let \( c \) be the largest such that \( (2c)^r < k/4 \). \( Y \) be the permutation on \( [m] \) such that \( Y(i) = (i + 1) \mod m \) for all \( i \in [m] \) and \( N \) be the identity permutation, i.e., \( N(i) = i \) for all \( i \in [m] \). Finally, let \( S = m, m + \left\lfloor \frac{m}{r} \right\rfloor, m + \left\lfloor \frac{2m}{r} \right\rfloor, \ldots, 2m \) as in the statement of Theorem 1.

We use \( \pi \) to construct an \( r \)-round \( (c, S) \)-protocol II as follows: Given as input \( \text{nest}(c, S) \)-layered graph \( G = (\bigsqcup_{i=0}^\ell V_i, \bigsqcup_{i=1}^\ell E_i) \), where Alice’s input is the edges \( E_i \) for even \( i \) and Bob’s input is the edges \( E_i \) for odd \( i \), the protocol II behaves as follows:

1. Alice and Bob add dummy vertices to all layers so that all of them have size \( 2m \). For all odd (resp. even) \( i \in [\ell] \), Alice (resp. Bob) adds dummy edges to \( E_i \) so that it is a perfect matching between \( V_{i-1} \) and \( V_i \).

2. Alice and Bob add \( k/4 - \ell - 1 \) new layers of vertices of size \( 2m \) each. For all \( \ell < k' < k/4 \), if \( k' \) is even, then Alice adds the identity matching between layer \( k' - 1 \) and layer \( k' \) to her set of edges. Otherwise, Bob adds the identity matching to his set of edges.

3. The players make a copy of the graph, denoted by \( \tilde{G} \), with layers \( \tilde{V}_0, \tilde{V}_1, \ldots, \tilde{V}_{k/4-1} \).

4. Bob adds the following edge to his input (see Figure 1):

   a. an identity matching (i.e., \( i \)-th vertex to \( i \)-th vertex) between the bottom \( (i.e., \), not the dummy vertices added in Item (1)) \( m \) vertices in \( V_{k/4-1} \) and the bottom \( m \) vertices \( \tilde{V}_0 \),

   b. an identity matching between the bottom \( m \) vertices in \( \tilde{V}_{k/4-1} \) and the top \( m \) vertices in \( V_0 \),

   c. an identity matching between the top \( m \) vertices in \( V_{k/4-1} \) and the top \( m \) vertices in \( \tilde{V}_{k/4-1} \),

   d. an identity matching between the top \( m \) vertices in \( \tilde{V}_0 \) and the bottom \( m \) vertices in \( V_0 \).

5. Alice and Bob run the protocol \( \pi \) on the new graph \( G_{\text{OMC}} \).

Suppose the matching from top \( m \) vertices of \( V_0 \) to top \( m \) vertices of \( V_\ell \) is \( \sigma \). Observe that if \( G \) has final matching \( \mathcal{Y} \), then the \( i \)-th vertex in bottom of \( V_0 \) connects to \( (i+1) \)-th vertex in bottom of \( V_{k/4-1} \), which connects to \( (i+1) \)-th vertex in bottom of \( \tilde{V}_0 \), and \( (i+2) \)-th vertex in bottom of \( \tilde{V}_{k/4-1} \). Then it connects to \( (i+2) \)-th vertex in the top of \( V_0 \), which connects to the \( \sigma(i+2) \)-th vertex in the top of \( V_{k/4-1} \), and then the \( \sigma(i+2) \)-th vertex in the top of \( \tilde{V}_{k/4-1} \). It then connects to the \( (i+2) \)-th vertex in the top of \( \tilde{V}_0 \), and finally, to the \( (i+2) \)-th vertex in the bottom of \( V_0 \). Therefore, since \( m \) is odd, all vertices are connected, and \( G_{\text{OMC}} \) is simply a Hamiltonian cycle. On the other hand, if \( G \) has final matching \( N \), one may verify similarly that \( G_{\text{OMC}} \) has \( n/k \) vertex-disjoint \( k \)-cycles. This means that \( \Pi \) has probability of success \( \geq (1/2 + (k/n)^2) \). On the other hand, \( \|\Pi\| = \|\pi\| \), and by Theorem 1,

\[
\|\Pi\| \geq 2^{-O(r)} \cdot m^{1-200/r}. 
\]
By the fact that \( \ell = (2c)^r \) and \( k/4 \leq (2(c + 1))^r \), we get \( c + 1 \geq \frac{1}{2}(k/4)^{1/r} \geq \frac{1}{8}k^{1/r} \). Next, if \( k^{1/r} < 10^4 \), the result is trivial. Assuming, otherwise, we have \( c \geq \frac{1}{16}k^{1/r} \), and thus,

\[
\|\pi\| \geq 2^{-\Theta(r)} \cdot m^{1-\frac{200}{c}} \\
\geq 2^{-\Theta(r)} \cdot m^{1-10^4k^{-1/r}}.
\]

This proves the theorem when \( k \) is a multiple of 4.

For even \( k \) that are not a multiple of 4, we make the same reduction for \( k' = k - 2 \) and \( m = n/k \). Then we split each vertex in the top of \( V_0 \) into a chain of length 3. The new graph has \( m \cdot k' + 2m = mk = n \) vertices. Moreover, if the graph is a Hamiltonian cycle before the split, then it remains a Hamiltonian cycle; if the graph was a disjoint union of \( k' \)-cycles, then the split adds 2 to the length of each cycle. Hence, this proves the theorem for all even \( k \).  

\[ \blacksquare \]

Remark 3.7. Observe that implicit in our proof above is a distribution \( D_{OMC} \) for the \( OMC_{n,k} \) problem that is “hard” for \( r \)-rounds protocols. It is notable that \( D_{OMC} \) can even be taken to be the uniform distribution that with probability \( \frac{1}{2} \), is the uniform distribution over all Hamiltonian cycles and with probability \( \frac{1}{2} \), is the uniform distribution on all graphs with \( (n/k) \) vertex-disjoint \( k \)-cycles provided that before running the protocol \( \pi \), Alice and Bob permute the identity of the vertices randomly (using public randomness).

Remark 3.8. By plugging in different pairs of \( Y \) and \( N \), we can prove lower bounds for distinguishing between different configurations of cycles in a graph. For example, by setting \( Y \) to the matching such that

\[
Y(i) = \begin{cases} 
  i - 3 & \text{if } 4 \mid i \\
  i + 1 & \text{otherwise}
\end{cases}
\]

and \( N(i) = i \), and assuming \( m = n/k \) is a multiple of four, the same argument proves that distinguishing between union of \( 2k \)-cycles and \( k \)-cycles is hard.
4 Lower Bound for One Round (Lemma 3.4)

We prove Lemma 3.4, restated below, in this section.

**Lemma 3.4 (restated).** Let \( S = \{ s_i \}_{i=1}^2 \) be an increasing sequence and \( c, C > 0 \) be integers. For all \( s \leq s_1 \), injections \( Y, N : [s] \to [s_1] \), and all randomized 1-round \((s,c,S)\)-protocols \( \Pi \) such that \( \|\Pi\| \leq C \), we have:

\[
\| \text{dist}_{G \sim \mathcal{D}_{s,c,S}}(\Pi(G)) - \text{dist}_{G \sim \mathcal{D}_{s,c,S}}^N(\Pi(G)) \|_{tvd} \leq 2s^2 \cdot \left( \frac{40C}{s_2 - s} \cdot \frac{C}{s_2} \right)^{c/6} = \Phi(s, c, s_2, C).
\]

**Proof.** To start, note that we can assume without loss of generality that \( s_1 = s_2 \). Indeed, if we show the lemma for \( s_1 = s_2 \), the lemma for \( s_1 < s_2 \) follows as any protocol for the latter case can be used to construct a protocol for the former case by Alice deleting \( s_2 - s_1 \) vertices from the last layer. We can also assume that \( \|\Pi\| = C \). In order to show the lemma, we show that if \( Y, N \) differ only in one coordinate, then:

\[
\| \text{dist}_{G \sim \mathcal{D}_{s,c,S}}(\Pi(G)) - \text{dist}_{G \sim \mathcal{D}_{s,c,S}}^N(\Pi(G)) \|_{tvd} \leq 2s^2 \cdot \left( \frac{40C}{s_2 - s} \cdot \frac{C}{s_2} \right)^{c/6}.
\]

The lemma then follows by triangle inequality as \( Y, N \) differ in at most \( s \) coordinates and \( s < s_2 \). Thus, there exists a sequence of injections \( Z_0, \ldots, Z_t \) for \( t \leq 2s \) such that \( Z_0 = Y, Z_t = N, Z_i \) and \( Z_{i+1} \) differ in exactly one coordinate.

Without loss of generality, assume that \( Y, N \) differ in the coordinate corresponding to vertex \( v_0 \in V_0 \). Denote the other \( s - 1 \) vertices by \( S = V_0 \setminus \{v_0\} \). Throughout this section, we use \( M_A \) to denote Alice’s input and \( M_B \) to denote Bob’s input. We abuse notation slightly and use \( \Pi \) to denote Alice’s message and \( \Pi^{out} \) to denote Bob’s output. The following lemma is the core of the proof. It bounds the \( \ell_2 \)-norm of the distribution of the image of \( v_0 \) after one message.

**Lemma 4.1 (\( \ell_2 \)-norm of the \( G_{0 \to 2c} \)-vector).** Let \( m > 0 \) and \( G \) be a graph sampled from \( \mathcal{D}_{1,c,(m,m)} \) with \( v_0 \) being the only vertex in the first layer. For any 1-round \((1,c,(m,m))\)-protocol \( \Pi \) with communication \( C \), we must have

\[
\mathbb{E}_{\Pi, M_B} \| G_{0 \to 2c}(v_0) \mid \Pi = \Pi, M_B = M_B \|^2 \leq \frac{1}{m} + \left( \frac{40C}{m} \right)^{c/3}.
\]

The lemma is proved in the next subsection. Here, we use it to prove Lemma 3.4. As Bob’s output is determined by his input and Alice’s message, we have:

\[
\| \text{dist}_{G \sim \mathcal{D}_{s,c,S}}(\Pi^{out}(G)) - \text{dist}_{G \sim \mathcal{D}_{s,c,S}}^N(\Pi^{out}(G)) \|_{tvd} \leq \| \Pi \|_{tvd} \| \Pi^{out}(G) - \Pi^{out}(G) \|_{tvd}
\]

\[
= \frac{1}{2} \cdot \sum_{\Pi, M_B} \mathbb{P}_{\mathcal{D}_{s,c,S}}(\Pi(G) = \Pi, M_B = M_B \mid G_{0 \to 2c} = Y) - \mathbb{P}_{\mathcal{D}_{s,c,S}}(\Pi(G) = \Pi, M_B = M_B \mid G_{0 \to 2c} = N).
\]

To continue, define \( W \) be the event that \( G_{0 \to 2c}(v) = Y(v) = N(v) \) for all \( v \neq v_0 \). Thus, the event \( G_{0 \to 2c} = Y \) is equivalent to \( W \) and \( G_{0 \to 2c}(v_0) = Y(v_0) \) and \( G_{0 \to 2c} = N \) is equivalent to \( W \) and \( G_{0 \to 2c}(v_0) = N(v_0) \). Also, for any \( a, b \in \{0\} \cup \{2c\} \), and any set \( S \subseteq V_a \), we denote by \( P_{[a,b]|S} \), the paths from \( S \) to the vertices in \( V_b \). By Bayes’s theorem, we have:

\[
\| \text{dist}_{G \sim \mathcal{D}_{s,c,S}}(\Pi^{out}(G)) - \text{dist}_{G \sim \mathcal{D}_{s,c,S}}^N(\Pi^{out}(G)) \|_{tvd}
\]
Next, we note that if \( G \) is a graph sampled from \( D \), then \( \| \) \( \sum_i v_i^2 \| \leq \sum_{i=1}^k v_i^2 \leq k \) \( \sum_i v_i^2 \leq k \).

To continue, note that the max of expectation is at most the expectation of the max. We have:

\[
\| \text{dist}_{G \sim D^Y_{s,c,S}}(\Pi_{\text{out}}(G)) - \text{dist}_{G \sim D^N_{s,c,S}}(\Pi_{\text{out}}(G)) \|_{\text{tvd}}
\leq s_2 \cdot \sqrt{\frac{E}{n,M_B,P_{[0,2c][S]|W}} \left[ \max_{v_2 \in V_{2c}} \left( \sum_{i=1}^k v_i \right) \right]}
\leq s_2 \cdot \sqrt{\frac{E}{n,M_B,P_{[0,2c][S]|W}} \left[ \sum_{i=1}^k v_i \right]}
\leq s_2 \cdot \left( \frac{40C}{s_2 - s + 1} \right)^{c/6}
\leq s_2 \cdot \left( \frac{40C}{s_2 - s} \right)^{c/6}.
\]

4.1 Block Matching of a Single Vertex After the First Round (Lemma 4.1)

In this subsection, we prove the \( \ell_2 \) bound on the matching of a single vertex after one message.

**Lemma 4.1 (repeated).** Let \( m > 0 \) and \( G \) be a graph sampled from \( D_{1,c,(m,m)} \) with \( v_0 \) being the only vertex in the first layer. For any 1-round \( (1,c,(m,m)) \)-protocol \( \Pi \) with communication \( C \), we must have

\[
\frac{E}{\Pi,M_B} \| G_{0 \to 2c}(v_0) \|_{\Pi = \Pi,M_B = M_B}^2 \leq \frac{1}{m} + \left( \frac{40C}{m} \right)^{c/3}.
\]

The rest of this subsection and the next one is then dedicated to the proof of Lemma 4.1.
Fix a message $\Pi$, and consider the distribution of Alice's input matchings conditioned on $\Pi = \Pi$. Let $v = (v_0, v_1, \ldots, v_{2c}) \in V_{\{0, \ldots, 2c\}}$ be a sequence of vertices (i.e., $v_i \in V_i$), and define:

$$p(v | \Pi) = \Pr[\forall i \in \{1, \ldots, c\}, G_{2i}(v_{2i-1}) = v_{2i} | \Pi],$$

that is, $p(v | \Pi)$ denotes the probability that the sequence $v$ is consistent with Alice's input conditioned on her sending the message $\Pi$.

Likewise, for any subset of Alice's matchings $S \subseteq \{1, \ldots, c\}$, we define $p_S$ as follows: For any $v_S \in V_{(2S-1)\cup 2S}$ (where $2S - 1 := \{2i - 1 | i \in S\}$ and similarly $2S := \{2i | i \in S\}$, i.e., the sets of endpoints of Alice's matchings in $S$)

$$p_S(v_S | \Pi) := \Pr[\forall i \in S, G_{2i}(v_{2i-1}) = v_{2i} | \Pi].$$

It is easy to verify the following claim (it simply calculates $p_S(v_S | \Pi)$ by going over all possible choices for edges in one extra matching of Alice which is not in $S$).

**Claim 4.2.** For any $i \notin S$, $v_{2i} \in V_{2i}$ and $v_S \in V_{(2S-1)\cup 2S}$, we have

$$p_S(v_S | \Pi) = \sum_{v_{2i-1} \in V_{2i-1}} p_{S \cup \{i\}}(v_S, v_{2i-1}, v_{2i} | \Pi).$$

Similarly, for any $v_{2i-1} \in V_{2i-1}$,

$$p_S(v_S | \Pi) = \sum_{v_{2i} \in V_{2i}} p_{S \cup \{i\}}(v_S, v_{2i-1}, v_{2i} | \Pi).$$

In the following lemma, we give a formula for the LHS of Lemma 4.1 in terms of $p_S$. We then use this formula in the next subsection to bound the LHS and conclude the proof.

**Lemma 4.3.** For any message $\Pi$,

$$\mathbb{E}_{M_B} ||G_{0 \rightarrow 2c}(v_0) | \Pi = \Pi, M_B = M_B||^2 = \frac{1}{m} + \frac{1}{m(m-1)c-1} \cdot \sum_{S \subseteq \{1, \ldots, c\}} \sum_{v_S \in V_{(2S-1)\cup 2S}} (-1)^{c-|S|} p_S(v_S | \Pi)^2.$$  

**Proof.** Note that, for all $v_{2c} \in V_{2c}$, we have:

$$\Pr[G_{0 \rightarrow 2c}(v_0) = v_{2c} | \Pi = \Pi, M_B = M_B] = \sum_{v \in \{v_0\} \times V_{1, \ldots, 2c-1} \times \{v_{2c}\}} p(v | \Pi) \cdot \prod_{i=1}^{c} 1_{G_{2i-1}(v_{2i-2}) = v_{2i-1}}.$$  

Since Alice and Bob's inputs are independent, we have

$$\mathbb{E}_{M_B} [\Pr[G_{0 \rightarrow 2c}(v_0) = v_{2c} | \Pi = \Pi, M_B = M_B]^2]$$

$$= \sum_{v, u \in \{v_0\} \times V_{1, \ldots, 2c-1} \times \{v_{2c}\}} p(v | \Pi)p(u | \Pi) \cdot \mathbb{E}_{M_B} \left[ \prod_{i=1}^{c} 1_{G_{2i-1}(v_{2i-2}) = v_{2i-1} \wedge G_{2i-1}(u_{2i-2}) = u_{2i-1}} \right]$$

$$= \sum_{v, u \in \{v_0\} \times V_{1, \ldots, 2c-1} \times \{v_{2c}\}} p(v | \Pi)p(u | \Pi) \cdot \prod_{i=1}^{c} \Pr[G_{2i-1}(v_{2i-2}) = v_{2i-1} \wedge G_{2i-1}(u_{2i-2}) = u_{2i-1}].$$
Observe that: (i) for \((v_{2i-2}, v_{2i-1}) = (u_{2i-2}, u_{2i-1})\),
\[
\Pr[G_{2i-1}(v_{2i-2}) = v_{2i-1} \land G_{2i-1}(u_{2i-2}) = u_{2i-1}] = \frac{1}{m};
\]
and (ii) for \(v_{2i-2} \neq u_{2i-2}\) and \(v_{2i-1} \neq u_{2i-1}\),
\[
\Pr[G_{2i-1}(v_{2i-2}) = v_{2i-1} \land G_{2i-1}(u_{2i-2}) = u_{2i-1}] = \frac{1}{m(m-1)};
\]
otherwise, it is equal to 0. Since \(u_0 = v_0\) and \(u_{2c} = v_2\), we have
\[
\sum_{v_{2c} \in V_{2c}} \mathbb{E} \left[ \Pr[G_0 \rightarrow 2c(v_0) = v_{2c} \mid \Pi = \Pi, M_B = M_B^2] \right] = \sum_{T \subseteq \{1, \ldots, c-1\}} \sum_{u, v \in \mathcal{R}^T, v_2 = u_{2c}, v_{2c+1} = u_{2c+1}} \frac{p(v \mid \Pi)p(u \mid \Pi)}{mc(m-1)^{|T|}}. \tag{8}
\]
Now, let us fix \(T, v_{2i} (= u_{2i})\) and \(v_{2i+1} (= u_{2i+1})\) for all \(i \notin T\), then take the sum over all \(v_{2i} \neq u_{2i}, v_{2i+1} \neq u_{2i+1}\) for all \(i \in T\). By inclusion-exclusion, we have
\[
\sum_{\forall i \in T, v_{2i} \neq u_{2i}, v_{2i+1} \neq u_{2i+1}} p(v \mid \Pi)p(u \mid \Pi) = \sum_{T' \subseteq 2T \cup (2T+1)} \sum_{\forall j \in (2T \cup (2T+1)) \setminus T', v_j = u_j \forall j \in T', v_j, u_j} (-1)^{|T'|-|T|} \cdot p(v \mid \Pi)p(u \mid \Pi)
\]
\[
= \sum_{T' \subseteq 2T \cup (2T+1)} (-1)^{|T'|} \cdot \sum_{\forall j \in (2T \cup (2T+1)) \setminus T', v_j} \left( \sum_{\forall j \in T', v_j} p(v \mid \Pi) \right)^2.
\]
Plug it into the RHS of (8), we get
\[
\sum_{T \subseteq \{1, \ldots, c-1\}} \sum_{\forall i \in T, v_{2i} \neq u_{2i}, v_{2i+1} \neq u_{2i+1}} \frac{p(v \mid \Pi)p(u \mid \Pi)}{mc(m-1)^{|T|}}
\]
\[
= \sum_{T \subseteq \{1, \ldots, c-1\}} \sum_{\forall i \in T, v_{2i} \neq u_{2i}, v_{2i+1} \neq u_{2i+1}} \frac{(-1)^{|T'|}}{mc(m-1)^{|T'|}} \cdot \sum_{\forall j \in (2T \cup (2T+1)) \setminus T', v_j} \left( \sum_{\forall j \in T', v_j} p(v \mid \Pi) \right)^2
\]
\[
= \sum_{T' \subseteq 2T \cup (2T+1)} (-1)^{|T'|} \cdot \sum_{\forall j \in \{1, \ldots, 2c\} \setminus T', v_j} \left( \sum_{\forall j \in T', v_j} p(v \mid \Pi) \right)^2,
\]
where \(|T'/2|\) denotes the set \(\{[i/2] : i \in T'\}\), and it is
\[
= \sum_{T' \subseteq \{2, \ldots, 2c-1\}} (-1)^{|T'|} \cdot \sum_{\forall j \in \{1, \ldots, 2c\} \setminus T', v_j} \left( \sum_{\forall j \in T', v_j} p(v \mid \Pi) \right)^2
\]
by Claim 4.2, it is
\[
\sum_{T' \subseteq \{2, \ldots, 2c-1\}} \frac{(-1)^{|T'|}}{m^{|T'|}} \sum_{\forall i \in \{1, \ldots, c\} \setminus |T'|/2|} (v | \Pi)^2 \cdot m^{|T'|}
\]
let \(S := \{1, \ldots, c\} \setminus |T'/2|\), it is
\[
= \frac{1}{m(m-1)^{c-1}} \cdot \sum_{S \subseteq \{1, \ldots, c\}} \sum_{\forall S \in V(2S-1), u \in S} p_S(v_S | \Pi)^2 \cdot \left( \sum_{T' \subseteq \{2, \ldots, 2c-1\} \setminus |T'/2| = \{1, \ldots, c\} \setminus S} \frac{(-m)^{|T'|}}{m^{|T'|}} \right).
\]
We claim that
\[
\sum_{T' \subseteq \{2, \ldots, 2c-1\} \setminus |T'/2| = \{1, \ldots, c\} \setminus S} \frac{(-m)^{|T'|}}{m^{|T'|}} = \begin{cases} (m-1)^{c-1} + (-1)^c & \text{if } S = \emptyset, \\ (-1)^{|S|} & \text{otherwise.} \end{cases} \tag{9}
\]
Since \(p_\emptyset = 1\), it is easy to verify that plugging Equation (9) proves the lemma.

To prove (9), define
\[
\alpha(m, c) := \sum_{T' \subseteq \{2, \ldots, 2c-1\} \setminus |T'/2| = \{1, \ldots, c\}} \frac{(-m)^{|T'|}}{m^{|T'|}},
\]
\[
\beta(m, c) := \sum_{T' \subseteq \{2, \ldots, 2c\} \setminus |T'/2| = \{1, \ldots, c\}} \frac{(-m)^{|T'|}}{m^{|T'|}},
\]
by symmetry, \(\beta(c)\) is also equal to
\[
\beta(m, c) = \sum_{T' \subseteq \{1, \ldots, 2c-1\} \setminus |T'/2| = \{1, \ldots, c\}} \frac{(-m)^{|T'|}}{m^{|T'|}},
\]
and
\[
\gamma(m, c) := \sum_{T' \subseteq \{1, \ldots, 2c\} \setminus |T'/2| = \{1, \ldots, c\}} \frac{(-m)^{|T'|}}{m^{|T'|}}.
\]
For \(S = \emptyset\), the left-hand-side of (9) is simply \(\alpha(m, c)\). Otherwise, suppose \(S = \{i_1, \ldots, i_s\}\), where \(1 \leq i_1 < i_2 < \cdots < i_s \leq c\). Then the sum in (9) only has \(T'\) such that \(2i_1 - 1, 2i_1, 2i_2 - 1, 2i_2, \ldots \notin T'\). These elements divide the set \(\{2, \ldots, c-1\}\) into \(s + 1\) chunks. Moreover, the constraint \(|T'/2| = \{1, \ldots, c\} \setminus S\) and the value \(\frac{(-m)^{|T'|}}{m^{|T'|}}\) are independent in all chunks. Therefore, the LHS is equal to
\[
\beta(m, i_1 - 1) \cdot \left( \prod_{l=1}^{s-1} \gamma(m, i_{l+1} - i_l - 1) \right) \cdot \beta(m, c - i_s).
\]
At last, we calculate \(\alpha, \beta\) and \(\gamma\). First, note \(\beta(m, c) = -\beta(m, c - 1)\). To see this, since \(|T'/2| = \{1, \ldots, c\}\), at least one of \(2c - 1\) and \(2c\) must be in \(T'\). There are three cases: (1) \(2c - 1, 2c \in T'\); (2) \(2c - 1 \in T'\) and \(2c \notin T'\); (3) \(2c - 1 \notin T'\) and \(2c \in T'\). Case (1) and case (2) cancel, since adding \(2c\) to the set increases both \(|T'|\) and \(|[T'/2]|\) by one, and the two cases have opposite signs by definition. Case (3) is equal to \(-\beta(m, c - 1)\). For the same reason,
we have \( \gamma(m, c) = -\gamma(m, c - 1) \). By the fact that \( \beta(m, 1) = -1 \) and \( \gamma(m, 1) = -1 \), we have \( \beta(m, c) = \gamma(m, c) = (\pm 1)^c \). This shows that the LHS when \( S \neq \emptyset \) is 
\[
(-1)^{i_1 - 1} \prod_{i=1}^{\ell} (-1)^{i_{i+1} - i_i} (-1)^{c_{i-1}} = (-1)^{c_{|S|}},
\]
proving (9) for \( S \neq \emptyset \).

On the other hand, for \( \alpha(m, c) \), \( 2c - 1 \) must be in \( T' \) (otherwise \( c \notin [T'/2] \)), and at least one of \( 2c - 3 \) and \( 2c - 2 \) must be in \( T' \). There are again three cases for \( 2c - 1, 2c - 2, 2c - 3 \):

1. \( 2c - 3, 2c - 2, 2c - 1 \in T' \); (2) \( 2c - 3, 2c - 2 \in T' \) and \( 2c - 2 \notin T' \); (3) \( 2c - 3 \notin T' \) and \( 2c - 2, 2c - 1 \in T' \). Case (1) is equal to \( m \cdot \alpha(m, c - 1) \); Case (2) is equal to \( -\alpha(m, c - 1) \); Case (3) is equal to \( m \cdot \beta(m, c - 2) \). Thus, we have the following recurrence:

\[
\alpha(m, c) = (m - 1)\alpha(m, c - 1) + m \cdot (\pm 1)^c,
\]
with \( \alpha(m, 1) = 0 \) and \( \alpha(m, 2) = m \). Expanding the recurrence, we get

\[
\alpha(m, c) = m(-1)^c + (m - 1)m(-1)^{c-1} + \cdots + (m - 1)^{c-2}m(-1)^2
\]
\[
= m(-1)^c \cdot \frac{1 - (1 - m)^{c-1}}{1 - (1 - m)}
\]
\[
= (m - 1)^{c-1} + (-1)^c.
\]
This completes the proof of the lemma.

The following lemma gives an upper bound on the expectation of each summand on the RHS of Lemma 4.3.

**Lemma 4.4.** For any \( S \subseteq \{1, \ldots, c\} \),

\[
\mathbb{E}_\Pi \left[ \sum_{v_S \in V_{(2S-1) \cup S}} p_S(v_S \mid \Pi)^2 \right] \leq \left( 4\sqrt{2(C + \log m)m + 6} \right)^{|S|},
\]
where we define \( \Gamma(m, C) := 4\sqrt{2(C + \log m)m < m/4} \).

We prove this lemma in the next subsection and for now, show how this lemma combined with Lemma 4.3 implies Lemma 4.1.

**Proof of Lemma 4.1.** We have,

\[
\text{LHS of Lemma 4.1} = \frac{1}{m} + \frac{1}{m(m - 1)^{c-1}} \cdot \sum_{S \subseteq \{1, \ldots, c\}} (-1)^{c_{|S|}} \cdot \mathbb{E}_\Pi \left[ \sum_{v_S \in V_{(2S-1) \cup S}} p_S(v_S \mid \Pi)^2 \right]
\]
\[
\leq \frac{1}{m} + \frac{1}{m(m - 1)^{c-1}} \cdot \sum_{S \subseteq \{1, \ldots, c\}} (\Gamma(m, C) + 6)^{|S|}
\]
(by the formula of Lemma 4.3)
\[
= \frac{1}{m} + \frac{1}{m(m - 1)^{c-1}} \cdot (\Gamma(m, C) + 7)^c
\]
(by the bound in Lemma 4.4 for each summand)
\[
\leq \frac{1}{m} + \left( \frac{40C}{m} \right)^{c/3},
\]
for sufficiently large \( m \) by the choice of \( \Gamma(m, C) := 4\sqrt{2(C + \log m)m} \). \( \blacksquare \)
4.2 Proof of Lemma 4.4

We are going to prove Lemma 4.4 inductively on the size of $S$, i.e., for all $S \subseteq [c]$:  

$$
\mathbb{E}_\Pi \left[ \sum_{v_S \in V(2^{S-1}) \cup 2^S} p_S(v_S | \Pi)^2 \right] \leq (\Gamma(m, C) + 6)^{|S|}, \tag{10}
$$

where $\Gamma(m, C) = 4\sqrt{2(C + \log m)m}$ as defined in Lemma 4.4.

The proof of this lemma builds heavily on our auxiliary information theory lemmas presented in Appendix A.1 and the reader may want to consult that section during this proof.

The base case is when $S = \emptyset$ and by definition, $p_\emptyset = 1$; hence, both sides are 1, and the inequality of (10) holds vacuously.

Let us now focus on the induction step. Fix any element $i \in S$, and let $S' := S \setminus \{i\}$. We have

$$
\mathbb{E}_\Pi \left[ \sum_{v_S \in V(2^{S-1}) \cup 2^S} p_S(v_S | \Pi)^2 \right] = \sum_{v_{S'} \in V(2^{S'-1}) \cup 2^{S'}} \mathbb{E}_\Pi \left[ \sum_{v_{2i-1} \in V_{2i-1} \cap v_{2i} \in V_{2i}} p_S(v_{S'}, v_{2i-1}, v_{2i} | \Pi)^2 \right] 
= \sum_{v_{S'} \in V(2^{S'-1}) \cup 2^{S'}} \mathbb{E}_\Pi \left[ p_{S'}(v_{S'} | \Pi)^2 \cdot \sum_{v_{2i-1} \in V_{2i-1} \cap v_{2i} \in V_{2i}} p_{\{i\}}(v_{2i-1}, v_{2i} | v_{S'}, \Pi)^2 \right]. \tag{11}
$$

In the following, for any $v_{S'}$, we define:

$$
\mathcal{B}(v_{S'}) := \{ \Pi \in \text{supp}(\Pi) \mid \text{Pr}[\Pi = \Pi | v_{S'}] \geq m^{-2}2^{-C} \}; \\
\mathcal{B} := \{ \Pi \in \text{supp}(\Pi) \mid \text{Pr}[\Pi = \Pi] \geq m^{-2}2^{-C} \}.
$$

In words, $\mathcal{B}(v_{S'})$ contains the messages that are “likely” conditioned on $v_{S'}$ and $\mathcal{B}$ contains the messages that are “likely” a-priori (without any conditioning).

We bound each summand in (11) for likely messages in the following claim. This is the place where we crucially use the fact that message of Alice is “small” (note that in the following, $\Psi$ would be small as we are conditioning on a likely message $\Pi$).

Claim 4.5. Fix $v_{S'}$. For any $\Pi \in \mathcal{B}(v_{S'})$:

$$
\sum_{v_{2i-1} \in V_{2i-1} \cap v_{2i} \in V_{2i}} p_{\{i\}}(v_{2i-1}, v_{2i} | v_{S'}, \Pi)^2 \leq \Psi(m, \Pi, G_{2i}),
$$

where we define $\Psi(m, \Pi, G_{2i}) := 4\sqrt{(\log m! - \mathbb{H}(G_{2i} | v_{S'}, \Pi = \Pi))m} + 4$.

Proof. By definition, $p_{\{i\}}(v_{2i-1}, v_{2i} | v_{S'}, \Pi)$ measures the probability of existence of any single edge in $G_{2i}$, thus,

$$
\sum_{v_{2i-1} \in V_{2i-1} \cap v_{2i} \in V_{2i}} p_{\{i\}}(v_{2i-1}, v_{2i} | v_{S'}, \Pi)^2 = \sum_{v_{2i-1} \in V_{2i-1}} \| G_{2i}(v_{2i-1}) | v_{S'}, \Pi = \Pi \|^2_2
$$
Now recall that $G_{2i} \mid v_{S'}$ is still a uniformly random matching and hence can be interpreted as a random permutation over $[m]$ (up to a renaming). Moreover, even conditioned on $\Pi = \Pi$, it has a high entropy since:

$$H(G_{2i} \mid \Pi = \Pi, v_{S'}) = \log m! - \mathbb{D}(G_{2i} \mid \Pi = \Pi, v_{S'}) \leq \log m - \log(1/\Pr(\Pi = \Pi \mid v_{S'}))$$ (by Fact A.5)

$$\geq \log m! - (C + 2\log m) \quad \text{(by the choice of } \Pi \in B(v_{S'}) \text{)}$$

$$> \log m! - m/8. \quad (13)$$

As such, $G_{2i} \mid v_{S'}, \Pi = \Pi$ is a high entropy random permutation of $[m]$ and we are interested in bounding the entropy of its average coordinate in (12). This is precisely the setting of Lemma A.13, which allows us to obtain,

$$(12) \leq 4\sqrt{(\log m! - H(G_{2i} \mid v_{S'}, \Pi = \Pi))m} + 4 = \Psi(m, \Pi, G_{2i}). \quad (by \ (13) \text{ and Lemma A.13)}$$

This concludes the proof of Claim 4.5.

We continue with the proof of Lemma 4.4. To bound (11), let us fix $v_{S'}$, and consider the expectation over $\Pi$:

$$\sum_{\Pi} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \sum_{v_{2i-1} \in V_{2i-1}} p_{i}(v_{2i-1}, v_{2i} \mid v_{S'}, \Pi)^2$$

$$= \sum_{\Pi \in B(v_{S'})} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \sum_{v_{2i-1} \in V_{2i-1}} p_{i}(v_{2i-1}, v_{2i} \mid v_{S'}, \Pi)^2$$

$$+ \sum_{\Pi \notin B(v_{S'})} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \sum_{v_{2i-1} \in V_{2i-1}} p_{i}(v_{2i-1}, v_{2i} \mid v_{S'}, \Pi)^2, \quad (14)$$

which by Claim 4.5 for the first term and the trivial upper bound for the second, is at most

$$\leq \sum_{\Pi \in B(v_{S'})} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \Psi(m, \Pi, G_{2i})$$

$$+ \sum_{\Pi \notin B(v_{S'})} m^{-2} \cdot 2^{-C} \cdot p_{S'}(v_{S'}) \cdot m^2 \quad \text{(where we define } p_{S'}(v_{S'}) := p_{S'}(v_{S'} \mid \emptyset))$$

$$\leq \left( \sum_{\Pi} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \Psi(m, \Pi, G_{2i}) \right) + p_{S'}(v_{S'}). \quad (15)$$

Next, we split the first term in (15) according to $\Pr(\Pi = \Pi)$, and it is equal to

$$= \sum_{\Pi \in B} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \Psi(m, \Pi, G_{2i}) + \sum_{\Pi \notin B} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \Psi(m, \Pi, G_{2i})$$
\[
\leq \sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \Psi(m, \Pi, G_{2i}) + \sum_{\Pi \notin \mathcal{B}} m^{2-2C} \cdot p_{S'}(v_{S'} \mid \Pi) \cdot m \log m
\]

\[
\leq \sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \Psi(m, \Pi, G_{2i}) + \sum_{\Pi \notin \mathcal{B}} 2^{-C} \cdot p_{S'}(v_{S'} \mid \Pi).
\]

(16)

We further simplify the first term in (16) in the following claim.

**Claim 4.6.** For the first term in (16),

\[
\sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \Psi(m, \Pi, G_{2i}) \leq (\Gamma(m, C) + 4) \cdot \sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2,
\]

for the parameter \(\Gamma(m, C) := 4\sqrt{2(C + \log m)}m\) defined in Lemma 4.4.

**Proof.** By the definition of \(\Psi\) in Claim 4.5, the LHS of the claim is equal to

\[
\sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2 \cdot \left(4\sqrt{(\log m! - H(G_{2i} \mid v_{S'}, \Pi = \Pi))m + 4}\right)
\]

which by Jensen’s inequality and the concavity of square root, is at most

\[
\leq \left(\sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2\right) \left(4 \cdot \sqrt{\left(\log m! - \sum_{\Pi \in \mathcal{B}} q(\Pi) \cdot H(G_{2i} \mid v_{S'}, \Pi = \Pi)\right)m + 4}\right),
\]

(17)

where we define \(q(\Pi) \propto \Pr(\Pi = \Pi)p_{S'}(v_{S'} \mid \Pi)^2 \propto \Pr(\Pi = \Pi)^{-1} \Pr(\Pi = \Pi \mid v_{S'})^2\) and \(\sum_{\Pi \in \mathcal{B}} q(\Pi) = 1\).

Notice the term \(\sum_{\Pi \in \mathcal{B}} q(\Pi) \cdot H(G_{2i} \mid v_{S'}, \Pi = \Pi)\) in above which is a weighted sum of entropy terms. In Lemma A.14, we give a simple auxiliary lemma that allows for lower bounding such weighted sum of entropy terms.

We apply Lemma A.14 for \(A\) being \((\Pi \mid v_{S'})\) and \(M\) being \((G_{2i} \mid v_{S'})\), and further set

\[
t = m!, \quad \ell := 2^C, \quad \alpha(\Pi) := \Pr(\Pi = \Pi \mid v_{S'}), \quad \beta(\Pi) := \Pr(\Pi = \Pi), \quad \epsilon := m^{2-2C}.
\]

Consequently,

\[
q(\Pi) = \frac{\beta(\Pi)^{-1} \alpha(\Pi)^2}{\sum_{\Pi' \in \mathcal{B}} \beta(\Pi')^{-1} \alpha(\Pi')^2}.
\]

Then since \(G_{2i}\) is a uniformly random matching conditioned on \(v_{S'}\), we can apply Lemma A.14 and have,

\[
\sum_{\Pi \in \mathcal{B}} q(\Pi) \cdot H(G_{2i} \mid v_{S'}, \Pi = \Pi) \geq \log m! - 2(C + \log m).
\]

Therefore,

\[
(17) \leq \left(\sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2\right) \left(4\sqrt{2m(C + \log m) + 4}\right)
\]

\[
= \left(\sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_{S'}(v_{S'} \mid \Pi)^2\right) (\Gamma(m, C) + 4),
\]

by the choice of \(\Gamma(m, C)\) as desired. \(\square\)
Finally, by Claim 4.6 and (16), (15), we obtain that for any fixed $v_{S'}$,
\[
\sum_{\Pi} \Pr(\Pi = \Pi) \cdot p_S'(v_{S'} | \Pi)^2 \cdot \sum_{v_{2i-1} \in V_{2i-1} \atop v_{2i} \in V_{2i}} p(\Pi') \cdot (v_{2i-1}, v_{2i} | v_{S'}, \Pi)^2
\leq p_S'(v_{S'}) + \sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_S'(v_{S'} | \Pi)^2 \cdot \Psi(m, \Pi, G_{2i}) \tag{by (15)}
\]
\[
\leq p_S'(v_{S'}) + \sum_{\Pi \notin \mathcal{B}} 2^{-C} \cdot p_S'(v_{S'} | \Pi) + \sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_S'(v_{S'} | \Pi)^2 \cdot \Psi(m, \Pi, G_{2i}) \tag{by (16)}
\]
\[
\leq p_S'(v_{S'}) + \sum_{\Pi \notin \mathcal{B}} 2^{-C} \cdot p_S'(v_{S'} | \Pi) + \sum_{\Pi \in \mathcal{B}} \Pr(\Pi = \Pi) \cdot p_S'(v_{S'} | \Pi)^2 \cdot (\Gamma(m, C) + 4) \tag{by Claim 4.6}
\]
\[
\leq p_S'(v_{S'}) + \sum_{\Pi} 2^{-C} \cdot p_S'(v_{S'} | \Pi) + \sum_{\Pi} \Pr(\Pi = \Pi) \cdot p_S'(v_{S'} | \Pi)^2 \cdot (\Gamma(m, C) + 4).
\]

By plugging these bounds in (11) and summing over all choices $v_{S'}$, we have
\[
(11) \leq \sum_{v_{S'} \in V_{(2s')-2s'}} \left( \Gamma(m, C) + 4 \right) \cdot \left( \sum_{\Pi} p(\Pi) \cdot p_S'(v_{S'} | \Pi)^2 \right) + \left( \sum_{\Pi} 2^{-C} \cdot p_S'(v_{S'} | \Pi) + p(v_{S'}) \right)
\]
\[
= \left( \Gamma(m, C) + 4 \right) \cdot \sum_{v_{S'} \in V_{(2s')-2s'}} \mathbb{E}_{\Pi} [p_S'(v_{S'} | \Pi)^2] + 2.
\]

That is,
\[
\mathbb{E}_{\Pi} \left[ \sum_{v_{S} \in V_{(2s)-2s}} p_S(v_{S} | \Pi)^2 \right] \leq \left( \Gamma(m, C) + 4 \right) \cdot \mathbb{E}_{\Pi} \left[ \sum_{v_{S'} \in V_{(2s')-2s'}} p_S'(v_{S'} | \Pi)^2 \right] + 2
\]
\[
\leq \left( \Gamma(m, C) + 4 \right) \cdot (\Gamma(m, C) + 6)^{|S'|} + 2 \tag{by the induction hypothesis for $S'$ in (10)}
\]
\[
\leq (\Gamma(m, C) + 6)^{|S'|},
\]
concluding the induction step, and the proof of Lemma 4.4.

5 Lower Bound For Multiple Rounds

In this section, we inductively extend the one round lower bound from Section 4 to many rounds. Specifically, we show that:

**Lemma 3.5 (rephrased).** Fix an increasing sequence $S$ and $c, C, \epsilon_0 > 0$. Assume that $|S| \geq 2$ and for all $s \leq s_2$, injections $Y', N': [s] \to [s_2]$, and all randomized $(|S| - 2)$-round $(s, c, S_{\geq 2})$-protocols $\Pi'$ such that $|\Pi'| \leq C$, we have:
\[
\|\text{dist}_{G \sim \mathcal{D}_{s,c,S_{\geq 2}}} (\Pi'(G')) - \text{dist}_{G \sim \mathcal{D}_{s,c,S_{\geq 2}}} (\Pi'(G'))\|_{\text{tvd}} \leq \epsilon_0.
\]

Then, for all $s \leq s_1$, injections $Y, N: [s] \to [s_1]$, and all randomized $(|S| - 1)$-round $(s, c, S)$-protocols $\Pi$ such that $|\Pi| \leq C$, we have:
\[
\|\text{dist}_{G \sim \mathcal{D}_{s,c,S}} (\Pi(G)) - \text{dist}_{G \sim \mathcal{D}_{s,c,S}} (\Pi(G))\|_{\text{tvd}} \leq 2c\epsilon_0 + \Phi(s, c, s_2, C),
\]

24
Proof. To start, note that we can assume without loss of generality the protocol \( \Pi \) is deterministic. Indeed, as a randomized protocol is simply a distribution over deterministic protocols, if we show the lemma for all deterministic protocols, we can use Lemma A.10 to also get the lemma for all randomized protocols.

Now, fix \( s \leq s' \), injections \( Y, N : [s] \to [s'] \), and a deterministic \((|S| - 1)\)-round \((s, c, S)\)-protocol \( \Pi \) such that \( ||\Pi|| \leq C \). We assume without loss of generality that \( |S| \) is even implying that Alice sends the last message in \( \Pi \). A symmetric argument, with Alice replaced by Bob, holds when \( |S| \) is odd.

Notation. Throughout this proof, we shall use \( S' = S \geq 2 \) and \( S = S \leq 2 \). Define the sequence \( T = \text{nest}(s, c, S) \) and \( \hat{T} = \text{nest}(s, c, \hat{S}) \). We use normal letters for objects associated with \( S \), letters with primes for objects associated with \( S' \), and letters with a “\(^\dagger\)” on top to denote objects associated with \( \hat{S} \). Thus, \( D'_{s,c,S} \) will denote \( D'_{s,c,S \geq 2} \). Let \( S, L \) denote the protocol where Alice and Bob first run the first \((l)\)-layered graph \( \hat{H} \) over the distribution \( \mathcal{L} \) as follows:

\[
\text{Exp}_{\hat{H}} := \forall i \in [2c] : \text{The subgraph } \{E_{(i-1)\ell} \}^{\ell}_{i=1} \text{ is nice and } G_{(i-1)\ell} = \hat{H}.
\]

For a \( \hat{T} \)-layered graph \( \hat{H} \), define \( \hat{H}_A = \{\hat{H}_i\}_{\text{even } i \in [\hat{\ell}]} \) and \( \hat{H}_B = \{\hat{H}_i\}_{\text{odd } i \in [\hat{\ell}]} \). Analogously (note that \( \hat{\ell} = 2c \)), for a \( T \)-layered graph \( H \), we define \( H_A = \{H_{(i-1)\ell} \}^{\ell}_{i=1} \) and \( H_B = \{H_{(i-1)\ell} \}^{\ell}_{i=1} \).

Let \( \Pi' \) denote the protocol where Alice and Bob first run the first \((|S| - 2)\)-rounds of \( \Pi \) and then Alice outputs the transcript of \( \Pi \), i.e., the messages exchanged in the first \((|S| - 2)\)-rounds and the message she would have sent in the last round (recall that Alice sends the last message in \( \Pi \)). Fix \( \hat{H} \) to be some canonical \( T \)-layered graph. We define the following notation for the edges of a \( T \)-layered graph \( \hat{H} = \left( \bigcup_{i=0}^{\hat{\ell}} V_i, \bigcup_{i=1}^{\hat{\ell}} E_i \right) \):

- For all \( i \in [2c] \), define \( L_i = (L_{i, \text{even}}, L_{i, \text{odd}}) \) where:

\[
L_{i, \text{even}} = \{E_{(i-1)\ell} \}^{\ell}_{i=1} \quad \text{and} \quad L_{i, \text{odd}} = \{E_{(i-1)\ell} \}^{\ell}_{i=1}.
\]

- Define \( L_{\text{pub}} = (L_{\text{pub, even}}, L_{\text{pub, odd}}) \) where:

\[
L_{\text{pub, even}} = \{L_{i, \text{even}}\}_{i=1}^{2c} \quad \text{and} \quad L_{\text{pub, odd}} = \{L_{i, \text{odd}}\}_{i=1}^{2c}.
\]

- Define:

\[
L_{\text{A-priv}} = \{L_{i, \text{even}}\}_{i=1}^{2c} \quad \text{and} \quad L_{\text{B-priv}} = \{L_{i, \text{odd}}\}_{i=1}^{2c}.
\]

Thus, \( H \) can be equivalently seen as the tuple \((L_{\text{pub}}, L_{\text{A-priv}}, L_{\text{B-priv}})\) and we use these representations interchangeably. We first claim that:

Lemma 5.1. For all \( \hat{T} \)-layered graphs \( \hat{H}^1, \hat{H}^2 \) such that \( \hat{H}^1_A = \hat{H}^2_A \), we have that:

\[
||\text{dist}_{H \sim \mathcal{L}|\text{Exp}_{\hat{H}^1}}(\Pi')(H), L_{\text{pub}}) - \text{dist}_{H \sim \mathcal{L}|\text{Exp}_{\hat{H}^2}}(\Pi')(H), L_{\text{pub}})||_{\text{tvd}} \leq \epsilon_0.
\]

Proof. To prove this lemma, we show that if the graphs \( \hat{H}^1, \hat{H}^2 \) differ in at most one layer, then:

\[
||\text{dist}_{H \sim \mathcal{L}|\text{Exp}_{\hat{H}^1}}(\Pi')(H), L_{\text{pub}}) - \text{dist}_{H \sim \mathcal{L}|\text{Exp}_{\hat{H}^2}}(\Pi')(H), L_{\text{pub}})||_{\text{tvd}} \leq \epsilon_0. \tag{18}
\]
Eq (18) implies the lemma using the triangle inequality as $\hat{H}^1, \hat{H}^2$ differ in at most $c$ layers. As the argument is symmetric for all layers, we assume without loss of generality that $\hat{H}^1$ and $\hat{H}^2$ differ only in the first layer.

Define $\hat{H}_i = \hat{H}^1_i = \hat{H}^2_i$ for all $i \neq 1 \in [2c]$ and consider the following ($|S| - 2$)-round $(s, c, S')$-protocol $\Pi'$:

**Algorithm 1** The protocol $\Pi'$.

**Input:** An $(s, c, S')$-almost nested block graph $G' = \left( \bigcup_{i'=0}^{c} V_{i'}', \bigcup_{i'=1}^{c} E_{i'}' \right)$. Alice gets the edges $E_i'$ for even $i'$ and Bob gets the edges $E_i'$ for odd $i$.

**Sample a $T$-layered graph $H$:**
1. Alice sets $L_{1, \text{even}} = \{E_{i'}'\}_{i'=1}^c$ and Bob sets $L_{1, \text{odd}} = \{E_{i'}'\}_{i'=1}^c$.
2. For all $i \in \{2, \cdots, 2c-1\}$, publicly sample edges $L_i$ from the distribution $D_{\hat{H}_i}$.
3. Publicly sample edges $L_{2c}$ from the distribution $D_{s_1}$.

**Simulate the protocol $\Pi'$ on $H$:**
4. Alice and Bob run the protocol $\Pi'$ on the graph $H$ and Alice outputs $\Pi'(H)$ and $L_{\text{pub}}$.

Next, note that when the input to $\Pi'$ is from $D_{s}$, then the graph $H$ sampled by $\Pi'$ distributes as $L \mid \text{Exp}_{\hat{H}_1}$. To see why, note that the edges $L_i$ are mutually independent for all $i \in [2c]$ in the distribution $L$ and conditioning on $\text{Exp}_{\hat{H}_1}$ does not change this fact. Thus, the distribution $L \mid \text{Exp}_{\hat{H}_1}$ is exactly the distribution sampled in Lines 1, 2, and 3, as desired.

Similarly, when the input to $\Pi'$ is from $D_{s}$, then the graph $H$ sampled by $\Pi'$ distributes as $L \mid \text{Exp}_{\hat{H}_2}$. We get:

$$\| \text{dist}_{H \sim L \mid \text{Exp}_{\hat{H}_1}} (\Pi'(H), L_{\text{pub}}) - \text{dist}_{H \sim L \mid \text{Exp}_{\hat{H}_2}} (\Pi'(H), L_{\text{pub}}) \|_{\text{tvd}}$$

$$\leq \| \text{dist}_{G' \sim D_{s}} (\Pi'(G')) - \text{dist}_{G' \sim D_{s}} (\Pi'(G')) \|_{\text{tvd}}$$

$$\leq \epsilon_0,$$

by the assumption in Lemma 3.5 and Eq (18) follows.

Next, we show the following property of $\Pi$.

**Lemma 5.2.** For the distribution $D_s$, it holds that:

$$\mathbb{I}(L_{B, \text{priv}}; G_{0\rightarrow t} | \Pi'(G), G_B, L_{\text{pub}}) = 0.$$

**Proof.** To start, note that conditioned on $G_B, L_{\text{pub}}$, the random variable $G_{0\rightarrow t}$ is determined by $L_{A, \text{priv}}$. Thus, we have from Item 7 of Fact A.1 that:

$$\mathbb{I}(L_{B, \text{priv}}; G_{0\rightarrow t} | \Pi'(G), G_B, L_{\text{pub}}) \leq \mathbb{I}(L_{B, \text{priv}}; L_{A, \text{priv}} | \Pi'(G), G_B, L_{\text{pub}}).$$

Next, recall that $\Pi$ is an $(|S| - 1)$-round communication protocol and let $\Pi_1(G), \cdots, \Pi_{|S|-1}(G)$ be the messages sent in $\Pi$ when run on input $G$. As $\Pi$ is deterministic, we have that for all odd (resp, even) $r \in [|S| - 1]$, the value $\Pi_r(G)$ is determined by Alice’s (resp, Bob’s) input and the values $\Pi_1(G), \cdots, \Pi_{r-1}(G)$. Using this fact repeatedly, we have:

$$\mathbb{I}(L_{B, \text{priv}}; G_{0\rightarrow t} | \Pi'(G), G_B, L_{\text{pub}}) \leq \mathbb{I}(L_{B, \text{priv}}; L_{A, \text{priv}} | \Pi_1(G), \cdots, \Pi_{|S|-1}(G), G_B, L_{\text{pub}})$$
\[ \leq I(L_{B \text{- priv}} : L_{A \text{- priv}} \mid G_B, L_{\text{pub}}) \quad \text{(Proposition A.3)} \\
\leq I(L_{B \text{- priv}}, L_{\text{pub, even}} : L_{A \text{- priv}}, L_{\text{pub, odd}} \mid G_B) \quad \text{(Fact A.1, Item 6)} \\
\leq I(\{L_i\}_{i \text{ odd}} \mid \{L_i\}_{i \text{ even}} \mid \{G_{(i-1)\ell' \rightarrow i\ell'}\}_{i \in [2c]}). \]

To finish, note that, conditioned on \( G_{(i-1)\ell' \rightarrow i\ell'} \) the edges in \( \{L_i\}_{\text{ odd}} \) and \( \{L_i\}_{\text{ even}} \) are independent, and therefore the right hand side is 0. \[ \square \]

Consider the following 1-round \((s, c, \hat{s})\)-protocol \( \hat{\Pi} \). Observe that in Lines 5, 6 we resample \( L_{i, \text{odd}} \). This is fine as our conditioning ensures that the two values are the same. Likewise for Lines 8, 9.

**Algorithm 2** The protocol \( \hat{\Pi} \).

**Input:** An \((s, c, \hat{s})\)-almost nested block graph \( \hat{G} = \left( \bigcup_{i=0}^{\hat{s}} \hat{V}_i; \bigcup_{i=1}^{\ell} \hat{E}_i \right) \). Alice gets the edges \( \hat{E}_i \) for even \( i \) and Bob gets the edges \( \hat{E}_i \) for odd \( i \).

1. Construct a \( \hat{T} \)-layered graph \( \hat{H} \) by setting \( \hat{H}_A = \hat{G}_A \) and \( \hat{H}_B = \hat{G}_B \).

**Sample a \( T \)-layered graph \( H \):**

2. Publicly sample \( L_{\text{pub}} \) uniformly at random.
3. for \( i \in [2c] \) do
4. if \( i \) is even then
5. If \( i = 2c \), Alice samples edges \( L_i \) from the distribution \( D^\hat{H}_i \mid L_{i, \text{odd}} = L_{i, \text{odd}} \).
6. If \( i < 2c \), Alice samples edges \( L_i \) from the distribution \( D^\hat{H}_i \mid L_{i, \text{odd}} = L_{i, \text{odd}} \).
7. else
8. If \( i = 1 \), Alice samples edges \( L_i \) from the distribution \( D^\hat{H}_i \mid L_{i, \text{even}} = L_{i, \text{even}} \).
9. If \( i > 1 \), Alice samples edges \( L_i \) from the distribution \( D^\hat{H}_i \mid L_{i, \text{even}} = L_{i, \text{even}} \).

**Simulate the protocol \( \Pi^\text{tr} \) on \( H \):**

10. Alice runs the protocol \( \Pi^\text{tr} \) on \( H \). Observe that she can do this without any communication as she knows all of \( H \). She sends the output \( \Pi^\text{tr}(H) \) to Bob.
11. Bob outputs \( (\hat{G}_B, \Pi^\text{tr}(H), L_{\text{pub}}) \).

Let \( \hat{H}^{\hat{\Pi}}(\hat{G}) \) be the graph constructed in \( \hat{\Pi} \) when the input is \( \hat{G} \). Also, define the random variable \( H^{\hat{\Pi}}(\hat{G}) \) be the graph \( H \) sampled in \( \hat{\Pi} \) when the input is \( \hat{G} \). Using Lemma 5.1, we conclude that:

**Lemma 5.3.** For all injections \( Z : [s] \rightarrow [s_1] \), we have that:

\[ \| \text{dist}_{G \sim D^\hat{\Pi}_1, G \sim \hat{L} \mid \text{Exp}_{\hat{G}}(\hat{G}, \Pi^\text{tr}(G), L_{\text{pub}})} - \text{dist}_{G \sim D^\hat{\Pi}_2, G \sim \hat{L} \mid \text{Exp}_{\hat{G}^{\hat{\Pi}}(\hat{G})}(\hat{G}, \Pi^\text{tr}(G), L_{\text{pub}})} \|_{\text{tvd}} \leq c\epsilon_0. \]

**Proof.** We have:

\[ \| \text{dist}_{G \sim D^\Pi_1, G \sim \hat{L} \mid \text{Exp}_{\hat{G}}(\hat{G}, \Pi^\text{tr}(G), L_{\text{pub}})} - \text{dist}_{G \sim D^\Pi_2, G \sim \hat{L} \mid \text{Exp}_{\hat{G}^{\hat{\Pi}}(\hat{G})}(\hat{G}, \Pi^\text{tr}(G), L_{\text{pub}})} \|_{\text{tvd}} \]

\[ \leq \mathbb{E}_{\hat{G}_B \sim D^\Pi_2} \left[ \| \text{dist}_{\hat{G}_B \sim D^\Pi_1, \hat{G}_B \sim \hat{L} \mid \text{Exp}_{\hat{G}}(\hat{G}, \Pi^\text{tr}(G), L_{\text{pub}})} - \text{dist}_{\hat{G}_B \sim D^\Pi_2, \hat{G}_B \sim \hat{L} \mid \text{Exp}_{\hat{G}^{\hat{\Pi}}(\hat{G})}(\hat{G}, \Pi^\text{tr}(G), L_{\text{pub}})} \|_{\text{tvd}} \right] \]

(Fact A.8)
A key property satisfied by the protocol $\hat{\Pi}$ is that:

**Lemma 5.4.** It holds for all $(s, c, \hat{S})$-almost nested block graphs $\hat{G}$ that:

$$\text{dist}(\hat{H}(\hat{G})) = \mathcal{L} \mid \text{Exp}_{\hat{H}(\hat{G})}.$$  

*Proof.* The proof is in two steps.

**The marginal distribution of $L_{\text{pub}}$ is identical.** First note, that the marginal distribution $L_{\text{pub}}$ in $\text{dist}(H(\hat{G}))$ is just uniform. We now consider the distribution $\mathcal{L} \mid \text{Exp}_{\hat{H}(\hat{G})}$. As the edges $L_i$ are mutually independent for all $i \in [2c]$ in the distribution $\mathcal{L}$ and conditioning on $\text{Exp}_{\hat{H}(\hat{G})}$ does not change this fact, the distribution $\mathcal{L} \mid \text{Exp}_{\hat{H}(\hat{G})}$ can be seen as simply sampling $L_i$ independently from the distribution $\mathcal{D}_{\text{pub}}(\hat{G})$ for $i \notin \{1, 2c\}$, from the distribution $\mathcal{D}_{\text{pub}}(\hat{G})$, when $i = 1$, and from the distribution $\mathcal{D}_{s_1,1}(\hat{G})$, when $i = 2c$. In particular, the marginal distribution of $L_{\text{pub}}$ is uniform.

**Conditioned on $L_{\text{pub}}$, the marginal distribution of $(L_{\text{A-priv}}, L_{\text{B-priv}})$ is identical.** The value $(L_{\text{A-priv}}, L_{\text{B-priv}})$ can be seen as the $2c$-tuple:

$$(L_{\text{A-priv}}, L_{\text{B-priv}}) = (\{L_{i, \text{even}}\}_{i=1}^{2c}, \{L_{i, \text{odd}}\}_{i=1}^{2c}).$$

We first show that the $2c$ coordinates are mutually independent in both distributions. In the distribution $\text{dist}(H(\hat{G})) \mid L_{\text{pub}}$, this is by Lines 5, 6, 8, and 9, while for the distribution $\mathcal{L} \mid \text{Exp}_{\hat{H}(\hat{G})}, L_{\text{pub}}$, this is because the coordinates $L_i$ are independent in $\mathcal{L}$ and conditioning on $\text{Exp}_{\hat{H}(\hat{G})}, L_{\text{pub}}$ does not change this fact.

To finish the proof, we fix a coordinate $i \in [2c]$ in this tuple and show that the marginal distribution of the $i^{th}$ coordinate is the same for both the distributions. Suppose $i = 1$ without loss of generality as the argument is the same for other values of $i$. The marginal distribution of $L_{i, \text{odd}}$ in the distribution $\text{dist}(H(\hat{G})) \mid L_{\text{pub}}$ is the corresponding marginal when the edges $L_i$ are sampled from $\mathcal{D}_{s_1,1}(\hat{G}) \mid L_{i, \text{even}} = L_{i, \text{even}}$. The latter is the same as the marginal distribution of $L_{i, \text{odd}}$ in the distribution $\mathcal{L} \mid \text{Exp}_{\hat{H}(\hat{G})}, L_{\text{pub}}$.

As the output of the protocol $\Pi$ is determined by the transcript and Bob’s input, we get:

$$\|\text{dist}_{\mathcal{G} \sim \mathcal{D}_s}(\Pi(G)) - \text{dist}_{\mathcal{G} \sim \mathcal{D}_s}(\Pi(G))\|_{\text{tv}}$$

$$\leq \|\text{dist}_{\mathcal{G} \sim \mathcal{D}_s}(G_B, \Pi^r(G), L_{\text{pub}}, L_{\text{B-priv}}) - \text{dist}_{\mathcal{G} \sim \mathcal{D}_s}(G_B, \Pi^r(G), L_{\text{pub}}, L_{\text{B-priv}})\|_{\text{tv}}$$

$$\leq \|\text{dist}_{\mathcal{G} \sim \mathcal{D}_s}(G_B, \Pi^r(G), L_{\text{pub}}) - \text{dist}_{\mathcal{G} \sim \mathcal{D}_s}(G_B, \Pi^r(G), L_{\text{pub}})\|_{\text{tv}}$$

$$+ \mathbb{E}_{(G_B, \Pi^r(G), L_{\text{pub}}) \sim \mathcal{D}_s} \left[\|\text{dist}_{\mathcal{G} \sim \mathcal{D}_s}(G_B, \Pi^r(G), L_{\text{pub}}) - \text{dist}_{\mathcal{G} \sim \mathcal{D}_s}(G_B, \Pi^r(G), L_{\text{pub}})\|_{\text{tv}}\right].$$

(Fact A.8)
To continue, note by Lemma 5.2 that, conditioned on Π_tr(G), G_\text{B}, L_{\text{pub}}, the random variable L_{\text{B-priv}} is independent of G_{\text{UB}}. Thus, the second term on the right hand side is 0. For the first term, observe that, for all injections Z : [s] → [s_1], sampling a graph G ∼ D_s Z is the same as first sampling a graph \( \hat{G} \sim D_s \) and then sampling G ∼ L | Exp_\hat{G}. We get:

\[
\| \text{dist}_{G \sim D_s}(\Pi(G)) - \text{dist}_{G \sim D_s}(\Pi(G)) \|_{\text{tvd}}
\]

\[
\leq \| \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) - \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) \|_{\text{tvd}}
\]

\[
\leq \| \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) - \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) \|_{\text{tvd}}
\]

\[
\leq \| \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) - \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) \|_{\text{tvd}}
\]

\[
+ \| \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) - \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) \|_{\text{tvd}}
\]

\[
+ \| \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) - \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) \|_{\text{tvd}}
\]

\[
(\text{Triangle Inequality})
\]

\[
\leq \| \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) - \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) \|_{\text{tvd}}
\]

\[
+ 2\varepsilon_0
\]

\[
(\text{Lemma 5.3})
\]

\[
\leq \| \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) - \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) \|_{\text{tvd}}
\]

\[
+ 2\varepsilon_0
\]

\[
(\text{Lemma 5.4})
\]

\[
\leq \| \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) - \text{dist}_{G \sim D_s}(\Pi_tr(G), L_{\text{pub}}) \|_{\text{tvd}}
\]

\[
+ 2\varepsilon_0 + 2\varepsilon_0 + \Phi(s, c, s_2, C).
\]

\[
(\text{Lemma 3.4})
\]

\[
\]

\[
\]

6 Graph Streaming Lower Bounds and Beyond

We now list some of the implications of our lower bound for OMC in Result 1 in proving streaming lower bounds for graph problems and beyond. Result 2 is formalized by these theorems.

We shall note that many of these reductions are simple (or sometimes even simpler) variants of prior reductions from BHH for these problems in the literature and we claim no particular novelty in their proofs (although in some cases such as maximum matching they even improve prior lower bounds for single-pass algorithms). Rather, we believe that these reductions showcase the role of OMC as a “multi-round version” of BHH for proving streaming lower bounds.

Throughout this section, for any integer \( p \geq 1 \) and \( \varepsilon \in (0, 1) \), we define:

\[
g(\varepsilon, p) := \beta \cdot \varepsilon^{1/2p - 1}
\]

for some absolute constant \( \beta > 0 \), depending only on the problem considered (taking \( \beta = 10^6 \) suffices for all the following applications).

Odd Cycles in the Reductions

Before we get to prove our streaming lower bounds, we prove a simple auxiliary lemma that is used in many of the following reductions. Recall that in the OMC_{n,k} problem, both \( n \) and \( k \) are even integers and this in particular implies that we are always working with even-length cycles. This is necessary in the definition of the problem as an odd-cycle cannot be partitioned into two matchings.
as input to players. But what if we are interested in the case when the input graph consists of many odd cycles or none at all? As it will become evident shortly, such lower bounds can be very helpful for proving streaming lower bounds for MAX-CUT, matching size, or bipartiteness testing. The following lemma establishes a simple reduction for this purpose\(^4\).

**Lemma 6.1.** Let \(n, k\) be even integers such that \(n/k\) is also even and sufficiently large. Let \(G\) be an input graph to \(\text{OMC}_{n,k}\). Suppose we sample \((n/k)\) edges from \(G\) uniformly at random and obtain a graph \(H\) from \(G\) by “stretching” each sampled edge into two edges and placing a new vertex between them. Then, (i) in the Yes case of \(\text{OMC}(G)\), \(H\) has no odd cycle, while (ii) in the No case of \(\text{OMC}(G)\), with probability at least \(9/10\), \(H\) has \(n/10k\) vertex-disjoint odd cycles of length \(k + 1\).

**Proof.** Let \(s = (n/k)\). The first case is trivial since \(H\) is now a Hamiltonian cycle of length \(n + s\) and both \(n\) and \(s\) are even.

We now analyze the second case. Let \(C_1, \ldots, C_{n/k}\) be the \((n/k)\) cycles in \(G\). For any \(C_i\), we define \(X_i \in \{0, 1\}\) which is one iff we stretch exactly one edge from \(C_i\) in \(H\). Notice that any cycle \(C_i\) with \(X_i = 1\) turns into an odd cycle in \(H\). For any \(i \neq j\), we have,

\[
\begin{align*}
\mathbb{E}[X_i] &= \frac{(n-k)}{s-1} \cdot \frac{(n-1)}{s-1} - \frac{(n-k)}{s-2} \cdot \frac{(n-2k)}{s-2} \\
\mathbb{Var}[X_i] &\leq \mathbb{E}[X_i^2] = \mathbb{E}[X_i] \\
\mathbb{Cov}[X_i, X_j] &= \frac{(n-k)}{s-1} \cdot \frac{(n-2k)}{s-2} - \frac{(n-k)}{s-1} \cdot \frac{(n-2k)}{s-2} \leq 0.
\end{align*}
\]

Define \(X := \sum_{i=1}^{n/k} X_i\) which lower bounds the number of odd cycles in \(H\). By the above calculation,

\[
\begin{align*}
\mathbb{E}[X] &\geq \frac{n}{2e \cdot k}; \\
\mathbb{Var}[X] &= \sum_i \mathbb{Var}[X_i] + \sum_{i \neq j} \mathbb{Cov}[X_i, X_j] \\
&\leq \sum_i \mathbb{E}[X_i] + \sum_{i \neq j} \mathbb{Cov}[X_i, X_j] \\
&\leq \frac{n}{2e \cdot k}.
\end{align*}
\]

As such, by Chebyshev’s inequality,

\[
\Pr\left( X \leq \frac{n}{10 \cdot k} \right) \leq \Pr\left( \left| X - \mathbb{E}[X] \right| \geq \frac{n}{5e \cdot k} \right) \leq \frac{\mathbb{Var}[X]}{(n/5e \cdot k)^2} \leq \frac{50 \cdot k}{n} \leq \frac{1}{10},
\]

which concludes the proof. \(\blacksquare\)

### 6.1 MAX-CUT

In the MAX-CUT problem, we are given an undirected graph \(G = (V, E)\) and our goal is to estimate the value of a maximum cut in \(G\), i.e., a partition of vertices into two parts that maximizes the number of crossing edges.

One can approximate MAX-CUT to within a factor of 2 in \(O(\log n)\) space (by counting the number of edges and dividing by two) and to \((1 + \varepsilon)\) in \(\tilde{O}(n/\varepsilon^2)\) space (by maintaining a cut

\(^4\)We could have alternatively used Theorem 1 to directly prove a lower bound for a slight variant of \(\text{OMC}_{n,k}\) for odd-values of \(k\) in which the input to players are no longer necessarily matchings—however, in the spirit of this section, we do this part using a reduction as well.
graphs and find the MAX-CUT of the sparsifier in exponential-time); see, e.g., [62]. A better two-pass algorithm for this problem on dense graphs is also presented in [21]. A direct reduction from BHH can prove that any (1+\(\varepsilon\))-approximation single-pass algorithm requires \(n^{1-O(\varepsilon)}\) space [62,67], and a series of work building on this approach [62–64] culminated in the optimal lower bound of \(\Omega(n)\) space for less-than-2 approximation in [64].

We prove the following theorem for MAX-CUT, formalizing part (i) of Result 2.

**Theorem 3.** Let \(p \geq 1\) be an integer and \(\varepsilon \in (0,1)\) be a parameter. Any \(p\)-pass streaming algorithm that outputs a (1+\(\varepsilon\))-approximation to the value of a maximum cut in undirected graphs with probability at least 2/3 requires \(\varepsilon \cdot 2^{-O(p)} n^{1-g(\varepsilon,p)}\) space where \(g\) is the function in Eq (19).

**Proof.** Suppose \(A\) is a \(p\)-pass \(S\)-space streaming algorithm for (1+\(\varepsilon\))-approximation of MAX-CUT. Consider any instance \(G\) of OMC\(_{n,k}\) for even integers \(n, k\) such that \((n/k) \leq 
\frac{1}{20\varepsilon}\) such that \(n/k\) is also even and is sufficiently large.

To solve OMC\((G)\), Alice and Bob use public coins to sample a graph \(H\) as in Lemma 6.1 (Alice has the original and stretched edges in \(M_A\) and Bob has the original and stretched edges in \(M_B\)). The players then run \(A\) on the stream obtained by appending edges of Bob to Alice’s edges. This can be done in 2\(p\) rounds and \(p \cdot S\) communication.

When OMC\((G)\) is a Yes-case, by Lemma 6.1, \(H\) has no odd cycle (is bipartite) and thus there is a cut that contain all its \(n + n/k\) edges. On the other hand, in the No-case of OMC\((G)\), again by Lemma 6.1, with probability 9/10, \(H\) has \(n/10\cdot k\) vertex-disjoint odd cycles. As any cut of this graph has to leave out at least one edge from an odd cycle, we obtain that the value of maximum cut in this case is at most \(n + 9n/10k\). By the choice of \(k\), we obtain that the value of MAX-CUT in the Yes-case is larger than the No-case by a factor of \((1 + \varepsilon)\). Hence, Alice and Bob can use the output of \(A\) to distinguish between the two cases.

To conclude, we obtain a 2\(p\) – 1-round \((2p - 1) \cdot S\)-communication that computes the correct answer to OMC\(_{n,k}\) with probability at least 2/3 – 1/10 > 1/2 + 1/m\(^2\). By the lower bound for OMC\(_{n,k}\) in Theorem 2, we obtain that

\[
S = 2^{-O(p)} \cdot (n/k)^{1-10^4 k^{-1/(2p-1)}} \geq \varepsilon 2^{-O(p)} \cdot n^{1-g(\varepsilon,p)}.
\]

as desired. 

**6.2 Maximum Matching Size**

In the maximum matching size estimation problem, we are given an undirected graph \(G = (V,E)\) and our goal is to estimate the size of a maximum matching in \(G\), i.e., the largest collection of vertex-disjoint edges of \(G\).

Maximum matching problem is one of the most studied problems in the graph streaming model. There is an active line of work on estimating matching size, in particular in planar and low arboricity graphs \([9,35,41,61,75]\). For single-pass algorithms, a reduction from BHH can prove an \(\Omega(\sqrt{n})\) space for better-than-(3/2) approximation in planar graphs \([41]\) which was improved to (1+\(\varepsilon\))-approximation in \(n^{1-O(\varepsilon)}\) space for low-arboricity graphs in \([26]\).

We prove the following theorem for matching size estimation, formalizing part (ii) of Result 2.

**Theorem 4.** Let \(p \geq 1\) be an integer and \(\varepsilon \in (0,1)\) be a parameter. Any \(p\)-pass streaming algorithm that outputs a (1+\(\varepsilon\))-approximation to the size of a maximum matching in planar graphs with probability at least 2/3 requires \(\varepsilon \cdot 2^{-O(p)} \cdot n^{1-g(\varepsilon,p)}\) space where \(g\) is the function in Eq (19).
Proof. The proof is almost identical to the proof of Theorem 3. Recall the reduction from $\text{OMC}_{n,k}$ in that theorem. When $\text{OMC}(G)$ is a Yes-case, $H$ has a perfect matching of size $\frac{(n + n/k)}{2}$ as each Hamiltonian cycle in a bipartite graph can be partitioned into two perfect matchings. On the other hand, when $\text{OMC}(G)$ is a No-case, any odd-cycle of $G$ on $k + 1$ vertices can only support a matching of size $\frac{k}{2}$. As such, the maximum matching size in this case is at most $\frac{(n + 9n/10k)}{2}$. The lower bound now holds verbatim as in Theorem 3.

Remark 6.2. We remark that the previous best lower bound of [26] for single-pass streaming algorithms on sparse graphs only hold for graphs of arboricity $\approx \frac{1}{\varepsilon}$ and not planar graphs. As such, our lower bound in Theorem 4 improves upon previous work even for single-pass algorithms.

6.3 Maximum Acyclic Subgraph

In the maximum acyclic subgraph problem, we are given a directed graph $G = (V,E)$ and our goal is to estimate the size of the largest acyclic subgraph in $G$, where the size of the subgraph is measured by its number of edges.

One can approximate this problem to within a factor of 2 in $O(\log n)$ space (by counting the number of edges and dividing by two). Under the Unique Games Conjecture [51], this is the best approximation ratio possible for this problem for poly-time algorithms but this does not imply a space lower bound in the streaming setting. This problem was studied in the streaming setting by [53, 54] and it was shown in [54] that a reduction from BHH implies an $\Omega(\sqrt{n})$-space lower bound for better-than-$(7/8)$ approximation of this problem in single-pass streams. We note that some related problems in directed graphs have also been studied recently in [28].

We prove the following theorem for this problem, formalizing part (iii) of Result 2.

**Theorem 5.** Let $p \geq 1$ be an integer and $\varepsilon \in (0, 1)$ be a parameter. Any $p$-pass streaming algorithm that outputs a $(1+\varepsilon)$-approximation to the number of edges in a largest acyclic subgraph of a directed graph with probability at least $2/3$ requires $\varepsilon 2^{-O(p)} \cdot n^{1 - g(\varepsilon, p)}$ space where $g$ is the function in Eq (19).

Proof. Suppose $A$ is a $p$-pass $S$-space streaming algorithm for $(1+\varepsilon)$-approximation of the maximum acyclic subgraph problem. Consider any instance $G$ of $\text{OMC}_{n,k}$ for even integers $n, k$ where $k \leq \frac{1}{4\varepsilon}$ such that $n/k$ is also even and is sufficiently large.

Unlike the proofs of Theorem 3 and Theorem 4, we do not need to introduce odd cycles in the graph $G$ (using Lemma 6.1). Instead we will direct the graph $G$ as follows. Let $L$ and $R$ be the bipartition of the bipartite graph $G$ which are known to both players (by the proof of Theorem 2). The players obtain the directed graph $H$ by Alice directing all her edges from $L$ to $R$ and Bob directing his from $R$ to $L$. The players then run $A$ on the stream obtained by appending edges of Bob to Alice’s edges. This can be done in $2p - 1$ rounds and $(2p - 1) \cdot S$ communication.

When $\text{OMC}(G)$ is a Yes-case, we can pick a subgraph of $H$ consisting of all but one edge and still have no cycle, hence the answer to the problem is $n - 1$ in this case. On the other hand, in the No-case of $\text{OMC}(G)$, we should leave out one edge from each cycle of the graph in the subgraph and thus the answer is $n - n/k$. By the choice of $k$, we obtain that the answer in the Yes-case is larger than the No-case by a factor of $(1 + \varepsilon)$. Hence, Alice and Bob can use the output of $A$ to distinguish between the two cases. The rest follows as before.

6.4 Minimum Spanning Tree

In the minimum spanning tree problem, we are given an undirected graph $G = (V,E)$ and with weight function $w : E \to \{1, 2, \ldots, W\}$ and our goal is to estimate the weight of the minimum
spanning tree (MST) in $G$.

There is a simple $O(n)$ space algorithm for computing an exact MST in a single pass [42]. More recently, [55] gave an $O(W \cdot n^{1-1.01\varepsilon/(W-1)})$ space algorithm for this problem, which achieves a polynomial saving in the space for any constant $\varepsilon, W > 0$. A reduction from BHH shows that single-pass $(1 + \varepsilon)$-approximation algorithms for this problem requires $\Omega(n^{1-\varepsilon/(W-1)})$ space [55].

We prove the following theorem for this problem, formalizing part (iv) of Result 2.

**Theorem 6.** Let $p \geq 1$ be an integer and $\varepsilon \in (0, 1)$ be a parameter. Any $p$-pass streaming algorithm that outputs a $(1 + \varepsilon)$-approximation to the weight of a minimum spanning tree in a graph with maximum weight $W$ with probability at least $2/3$ requires $\varepsilon W^{-2} - O(p) \cdot n^{1-\varepsilon(W/p)}$ space where $g$ is the function in Eq (19) (notice that the first argument of $g$ is $\varepsilon/W$ and not merely $\varepsilon$).

*Proof.* Suppose $A$ is a $p$-pass $S$-space streaming algorithm for $(1 + \varepsilon)$-approximation of the minimum weight spanning tree problem. Consider any instance $G$ of OMC$_{n,k}$ for even integers $n, k$ where $k \leq \frac{W}{4\varepsilon}$ such that $n/k$ is also even and is sufficiently large.

Given an instance $G$ of OMC$_{n,k}$, Alice and Bob create a graph $H$ as follows: $H$ contains $G$ as a subgraph with weight one on all edges. Moreover, $H$ has one additional vertex $s$ which is connected to all vertices of $G$ with weight $W$ except for one vertex $\hat{s} \in G$ where $s$ is connected to it with weight 1 instead (we give all latter edges to Bob). The players then run $A$ on the stream obtained by appending edges of Bob to Alice’s edges. This can be done in $(2p - 1)$ rounds and $(2p - 1) \cdot S$ communication.

When OMC($G$) is a Yes-case, a minimum spanning tree of $G$ has weight $n$ by picking the $n - 1$ edges of the Hamiltonian cycle plus the edge $(s, \hat{s})$, all of weight one. On the other hand, in the No-case of OMC($G$), we need to pick at least $n/k - 1$ edges of weight $W$ to connect $s$ to any cycle of $G$ that does not contain $\hat{s}$, as those cycles are not connected in $G$; the remainder of the edges can then be of weight 1 again (recall that $H$ has $n + 1$ vertices and so its MST needs $n$ edges). As such, in this case, the weight of MST is at least $n + (n/k - 1) \cdot (W - 1)$.

By the choice of $k$, we obtain that the answer in the Yes-case is smaller than the No-case by a factor of $(1 + \varepsilon)$. Hence, Alice and Bob can use the output of $A$ to solve OMC. The rest again follows as before (just notice that $k \approx W/\varepsilon$ and not $1/\varepsilon$ unlike the previous examples).  

### 6.5 Property Testing of Connectivity, Bipartiteness, and Cycle-freeness

Given a parameter $\varepsilon \in (0, 1)$, an $\varepsilon$-property tester algorithm for a property $P$ needs to decide whether $G$ has the property $P$ or is $\varepsilon$-far from having $P$, defined as follows:

- **Connectivity:** A graph is $\varepsilon$-far from being connected if we need to insert $\varepsilon \cdot n$ more edges to make it connected;

- **Bipartiteness:** A graph is $\varepsilon$-far from being bipartite if we need to delete $\varepsilon \cdot n$ of its edges to make it bipartite;

- **Cycle-freeness:** A graph is $\varepsilon$-far from being cycle-free if we need to delete $\varepsilon \cdot n$ of its edges to remove all its cycle.

Traditionally, these problems have been studied extensively in the query complexity model and for sublinear-time algorithms. However, starting from the pioneering work of [55], these problems have been receiving increasing attention in the streaming literature as well [37, 55, 77, 81].
particular, [55] gave single-pass streaming algorithms for these problems with $\widetilde{O}(n^{1-\text{poly}(\varepsilon)})$ space (for bipartiteness testing, the input graph is assumed to be planar). Moreover, a reduction from BHH implies that all these problems require $n^{1-O(\varepsilon)}$ space in a single-pass [55].

We prove the following theorem for streaming property testing, formalizing part (iv) of Result 2.

**Theorem 7.** Let $p \geq 1$ be an integer and $\varepsilon \in (0,1)$ be a parameter. Any $p$-pass streaming algorithm for $\varepsilon$-property testing of connectivity, bipartiteness, or cycle-freeness in planar graphs, with probability at least $2/3$ requires $\varepsilon \cdot 2^{-O(p)} \cdot n^{1-g(\varepsilon,p)}$ space where $g$ is the function in Eq (19).

**Proof.** The proofs of these parts are all simple corollaries of the above reductions from OMC.

For connectivity, we already discussed the reduction in Section 1.2. Basically, for $k \leq \frac{1}{2\varepsilon}$, the Yes-cases of OMC$_{n,k}$ are Hamiltonian cycles and hence connected, while the No-cases are disjoint-union of $> \varepsilon \cdot n$ cycles and hence need at least $\varepsilon \cdot n$ edge-insertions to become connected. Thus, property testing algorithms for connectivity can also solve OMC$_{n,k}$.

For bipartiteness, we rely on our proof of Theorem 3. The lower bound for MAX-CUT involved either bipartite graphs, or graphs where any cut misses $\varepsilon \cdot n$ edges of the graph – this makes the latter graphs $\varepsilon$-far from being bipartite and the proof follows similar to Theorem 3.

For cycle-freeness, we perform another reduction from OMC$_{n,k}$ for $k \leq \frac{1}{2\varepsilon}$. Given an instance $G$ of OMC$_{n,k}$, one of the players, say, Alice for concreteness, remove any one arbitrary edge from her input to obtain the graph $H$. Now, in the Yes-case of OMC, $H$ is a cycle-free graph as we removed one edge from a Hamiltonian cycle, while in the No-case, $H$ still consists of $n/k-1 > \varepsilon \cdot n$ vertex-disjoint cycle is thus $\varepsilon$-far from being cycle-free. The lower bound follows.

### 6.6 Matrix Rank and Schatten Norms

For any $q \geq 0$, the Schatten $q$-norm of an $n$-by-$n$ matrix $A$ is the $\ell_q$-norm of the vector of the singular values, i.e., $(\sum_{i=1}^{n} \sigma_i(A)^q)^{1/q}$ where $\sigma_1(A), \ldots, \sigma_n(A)$ are the singular values of $A$ (we used the notation $q$ instead of $p$ in expressing the norms to avoid ambiguity with number of passes of the streaming algorithm). In particular, the case of $q = 0$ corresponds to the rank of matrix $A$, $q = 1$ is the nuclear (or trace) norm, and $q = 2$ corresponds to the Frobenius norm.

Streaming algorithms for computing Schatten $q$-norms of matrices (particularly sparse matrices) whose entries are updated in a stream have gain considerable attention lately [9,22,23,26,32,70]. As this topic is somewhat beyond the scope of our paper on graph streams, we refer the interested reader to [22,23,70] for an overview of single-pass and multi-pass streaming algorithms for this problem. But we note that reductions from BHH have been used to establish $n^{1-O(\varepsilon)}$ space lower bounds for single-pass algorithms that $(1+\varepsilon)$-approximate rank [26], or any value of $q \in [0, +\infty)$ which is not an even integer [22,70]. We remark that [22,70] also have $\Omega(n^{1-4/q})$ space lower bounds for even values of $q$ but this one requires a reduction from set disjointness and not BHH and is weaker than the bounds obtained when $q$ is not an even integer (similar to all the other BHH lower bounds, extending the lower bounds for values of $q$ which are not even integers to multi-pass algorithms was left open in [22,70]).

**Theorem 8.** Let $q \in [0, +\infty) \setminus 2Z$, $p \geq 1$ be an integer and $\varepsilon \in (0,1)$ be a parameter. Any $p$-pass streaming algorithm for $(1+\varepsilon)$-approximation of Schatten $q$-norms in sparse matrices, with probability at least $2/3$ requires $\varepsilon \cdot 2^{-O(p)} \cdot n^{1-g(\varepsilon,p)}$ space where $g$ is the function in Eq (19).

**Proof.** The proof for matrix rank, i.e., the $q = 0$ case, follows from Theorem 4, and the standard equivalence between estimating matching size and computing the rank of the Tutte matrix [84] with randomly chosen entries established in [71] (see [26] for this reduction in the streaming model).
For other values of \( q \), we rely on the following lemma established by \([22, \text{Lemma 5.2}]\).

**Lemma 6.3** (Lemma 5.2 in \([22]\)). Suppose that \( t \geq 2 \) is an integer and \( q \in (0, +\infty) \setminus 2\mathbb{Z} \). Let \( G \) be an undirected \( 2 \)-regular graph consisting of either (i) vertex-disjoint \((2t + 1)\)-cycles or (ii) vertex-disjoint \((4t + 2)\)-cycles. Then, the Schatten \( q \)-norm of the Laplacian matrix \( L_G \) associated with \( G \) differs by a constant factor depending only on \( t \) and \( q \) between the two cases.

Note that two possible cases of the graph \( G \) in Lemma 6.3 correspond to the cases of the \( k \)-vs-2\( k \) communication (which is a “harder” variant of \( \text{OMC}_{n,k} \)) instead of directly mapping to \( \text{OMC}_{n,k} \). As such, for this proof, we instead use our more general communication lower bound that holds for distinguishing any two profiles of \( k \)-cycles (even when \( k \) is odd), namely, Remark 3.8.\(^5\) As the players in the communication problem can easily form the Laplacian of their input graph and run the Schatten norm streaming algorithm, the lower bound follows as before.  

### 6.7 Sorting-by-Block-Interchange

Given a string \( x = (x_1, \ldots, x_n) \) representing a permutation on \( n \) elements (i.e., \( x_1, \ldots, x_n \) are distinct integers in \([n]\)), we define the **block-interchange** operation on \( x \), denoted by \( r(i,j,k,l) \) where \( 1 \leq i \leq j < k \leq l \leq n \), as transforming \( x \) to the string

\[
x' = (x_{[1,i-1]}, x_{[k,l]}, x_{[j+1,k-1]}, x_{[i,j]}, x_{[l+1,n]}),
\]

where \( x_{[a,b]} = x_a, x_{a+1}, \ldots, x_b \) (basically, \( x' \) is obtained from \( x \) by replacing the position of two blocks of sub-strings of \( x \) with each other). We define the value of the **sorting-by-block-interchange** function on input \( x \) to be the minimum number of block-interchanges that are required to sort \( x \) in an increasing order (i.e., get \((1,2,\ldots,n)\)).

In the streaming model, the entries of \( x \) are arriving one by one in the stream and our goal is to output the value of sorting-by-block-interchange for \( x \). This problem is one of the several string processing problems studied by \([85]\) who proved an \( n^{1-O(\varepsilon)} \) space lower bound for it using a reduction from \( \text{BHH} \).

We can now prove the following lower bound for this problem.

**Theorem 9.** Let \( p \geq 1 \) be an integer and \( \varepsilon \in (0,1) \) be a parameter. Any \( p \)-pass streaming algorithm that outputs a \((1+\varepsilon)\)-approximation of the value the sorting-by-block-interchange for a length-\( n \) string with probability at least \( 2/3 \) requires \( \varepsilon \cdot 2^{-O(p)} \cdot n^{1-g(\varepsilon,p)} \) space where \( g \) is the function in Eq (19).

**Proof.** The proof of this theorem follows directly from \([85, \text{Theorem 2.2}]\) who gave a reduction from the \( k \)-vs-2\( k \) cycle problem on graphs with \( 4n \) vertices and \( k = O(1/\varepsilon) \), and our lower bound for the \( k \)-vs-2\( k \) cycle problem in Remark 3.8 exactly as in Theorem 8.  

This concludes the list of all the lower bounds mentioned in Result 2.

---

\(^5\)It seems very plausible to show that the Laplacian of the graphs in the two cases of \( \text{OMC}_{n,k} \) also differ by a constant factor using a similar argument to \([22]\) and so one could still use \( \text{OMC} \) directly; however, we did not pursue this direction in the paper.
References


40


Appendix

A Tools from Information Theory

We briefly list the definitions and tools from information theory that we use in this paper.

- We denote the Shannon Entropy of a random variable \( A \) by \( \mathbb{H}(A) \), which is defined as:

\[
\mathbb{H}(A) := \sum_{A \in \text{supp}(A)} \mu(A) \cdot \log \left( \frac{1}{\mu(A)} \right).
\]  

(20)

- The conditional entropy of \( A \) conditioned on \( B \) is denoted by \( \mathbb{H}(A \mid B) \) and defined as:

\[
\mathbb{H}(A \mid B) := \mathbb{E}_{B \sim B} \left[ \mathbb{H}(A \mid B = B) \right],
\]  

(21)

where \( \mathbb{H}(A \mid B = B) \) is defined in a standard way by using the distribution of \( A \) conditioned on the event \( B = B \) in Eq (20).

- The mutual information of \( A \) and \( B \) is denoted by \( \mathbb{I}(A; B) \) and is defined as:

\[
\mathbb{I}(A; B) := \mathbb{H}(A) - \mathbb{H}(A \mid B) = \mathbb{H}(B) - \mathbb{H}(B \mid A).
\]  

(22)

- The conditional mutual information \( \mathbb{I}(A; B \mid C) \) is defined similarly as:

\[
\mathbb{I}(A; B \mid C) := \mathbb{H}(A \mid C) - \mathbb{H}(A \mid B, C) = \mathbb{H}(B) - \mathbb{H}(B \mid A, C).
\]  

(23)

Standard Properties of Entropy and Mutual Information

We shall use the following basic properties of entropy and mutual information throughout. Proofs of these properties mostly follow from convexity of the entropy function and Jensen’s inequality and can be found in [36, Chapter 2].

**Fact A.1.** Let \( A, B, C, \) and \( D \) be four (possibly correlated) random variables.

1. \( 0 \leq \mathbb{H}(A) \leq \log |\text{supp}(A)| \). The right equality holds iff \( \text{dist}(A) \) is uniform.

2. \( \mathbb{I}(A; B) \geq 0 \). The equality holds iff \( A \) and \( B \) are independent.

3. Conditioning on a random variable reduces entropy: \( \mathbb{H}(A \mid B, C) \leq \mathbb{H}(A \mid B) \). The equality holds iff \( A \perp C \mid B \).

4. Subadditivity of entropy: \( \mathbb{H}(A, B \mid C) \leq \mathbb{H}(A \mid C) + \mathbb{H}(B \mid C) \).

5. Chain rule for entropy: \( \mathbb{H}(A, B \mid C) = \mathbb{H}(A \mid C) + \mathbb{H}(B \mid C, A) \).

6. Chain rule for mutual information: \( \mathbb{I}(A, B; C \mid D) = \mathbb{I}(A; C \mid D) + \mathbb{I}(B; C \mid A, D) \).

7. Data processing inequality: *suppose \( f(A) \) is a deterministic function of \( A \), then*

\[
\mathbb{I}(f(A); B \mid C) \leq \mathbb{I}(A; B \mid C).
\]

We also use the following two standard propositions.
Proposition A.2. For random variables $A, B, C, D$, if $A \perp D | C$, then,
$$I(A; B | C) \leq I(A; B | C, D).$$

Proof. Since $A$ and $D$ are independent conditioned on $C$, by Fact A.1-(3), $\mathbb{H}(A | C) = \mathbb{H}(A | C, D)$ and $\mathbb{H}(A | C, B, D) \geq \mathbb{H}(A | C, B)$. We have,
$$I(A; B | C) = \mathbb{H}(A | C) - \mathbb{H}(A | B, C) = \mathbb{H}(A | C, D) - \mathbb{H}(A | C, B)$$
$$\leq \mathbb{H}(A | C, D) - \mathbb{H}(A | C, B, D) = I(A; B | C, D). \qed$$

Proposition A.3. For random variables $A, B, C, D$, if $A \perp D | B, C$, then,
$$I(A; B | C) \geq I(A; B | C, D).$$

Proof. Since $A \perp D | B, C$, by Fact A.1-(3), $\mathbb{H}(A | B, C) = \mathbb{H}(A | B, C, D)$. Moreover, since conditioning can only reduce the entropy (again by Fact A.1-(3)),
$$I(A; B | C) = \mathbb{H}(A | B, C) - \mathbb{H}(A | B, C, D) = \mathbb{H}(A | D, C) - \mathbb{H}(A | B, C, D) = I(A; B | C, D). \qed$$

Measures of Distance Between Distributions

We use two main measures of distance (or divergence) between distributions, namely the Kullback-Leibler divergence (KL-divergence) and the total variation distance.

KL-divergence. For two distributions $\mu$ and $\nu$ over the same probability space, the Kullback-Leibler divergence between $\mu$ and $\nu$ is denoted by $D(\mu || \nu)$ and defined as:
$$D(\mu || \nu) := \mathbb{E}_{a \sim \mu} \left[ \log \frac{\mu(a)}{\nu(a)} \right].$$

We have the following relation between mutual information and KL-divergence.

Fact A.4. For random variables $A, B, C \sim \mu$,
$$I(A; B | C) = \mathbb{E}_{(b,c) \sim (B,C)} \left[ D(\text{dist}_\mu(A | B = b, C = c) || \text{dist}_\mu(A | C = c)) \right].$$

One can write the entropy of a random variable as its KL-divergence from the uniform distribution.

Fact A.5. Let $A$ be any random variable and $\mathcal{U}$ be the uniform distribution on $\text{supp}(A)$. Then,
$$\mathbb{H}(A) = \log |\text{supp}(A)| - D(A || \mathcal{U}).$$

Finally, we also have the following property.

Fact A.6. Let $A$ be any random variable and $E$ be any event in the same probability space. Then,
$$D(A | E || A) \leq \log \frac{1}{\Pr(E)}.$$
**Total variation distance.** We denote the total variation distance between two distributions \( \mu \) and \( \nu \) on the same support \( \Omega \) by \( \| \mu - \nu \|_{\text{tvd}} \), defined as:

\[
\| \mu - \nu \|_{\text{tvd}} := \max_{\Omega' \subseteq \Omega} (\mu(\Omega') - \nu(\Omega')) = \frac{1}{2} \cdot \sum_{x \in \Omega} |\mu(x) - \nu(x)|. \tag{25}
\]

We use the following basic properties of total variation distance.

**Fact A.7.** Suppose \( \mu \) and \( \nu \) are two distributions for \( E \), then, \( \mu(E) \leq \nu(E) + \| \mu - \nu \|_{\text{tvd}} \).

We also have the following bound on the total variation distance of joint variables.

**Fact A.8.** For any two distributions \( p \) and \( q \) of an \( n \)-tuple \( (X_1, X_2, \ldots, X_n) \), we must have

\[
\| \text{dist}_p(X_1, \ldots, X_n) - \text{dist}_q(X_1, \ldots, X_n) \|_{\text{tvd}} \leq \sum_{i=1}^{n} \mathbb{E}_{(X_1, \ldots, X_{i-1}) \sim p} \| \text{dist}_p(X_i \mid X_1, \ldots, X_{i-1}) - \text{dist}_q(X_i \mid X_1, \ldots, X_{i-1}) \|_{\text{tvd}}. \]

Finally, the following Pinsker’s inequality bounds the total variation distance between two distributions based on their KL-divergence,

**Fact A.9 (Pinsker’s inequality).** For any distributions \( \mu \) and \( \nu \), \( \| \mu - \nu \|_{\text{tvd}} \leq \sqrt{\frac{1}{2} \cdot D(\mu \mid \mid \nu)} \).

### A.1 Auxiliary Information Theory Lemmas

This section includes some additional information theory lemmas that we prove in this paper.

**Total variation distance of convex combinations of distributions.** The following lemma is a consequence of the triangle inequality for the total variation distance.

**Lemma A.10.** Let \( \Omega \) and \( I \) be non-empty sets and for all \( i \in I \), distributions \( \mu_i \) and \( \nu_i \) supported on \( \Omega \) be given. For every distribution \( \mathcal{I} \) supported on \( I \), we have

\[
\| \mathbb{E}_{i \sim \mathcal{I}} [\mu_i] - \mathbb{E}_{i \sim \mathcal{I}} [\nu_i] \|_{\text{tvd}} \leq \mathbb{E}_{i \sim \mathcal{I}} [\| \mu_i - \nu_i \|_{\text{tvd}}].
\]

**Proof.** We have:

\[
\| \mathbb{E}_{i \sim \mathcal{I}} [\mu_i] - \mathbb{E}_{i \sim \mathcal{I}} [\nu_i] \|_{\text{tvd}} = \frac{1}{2} \cdot \sum_{x \in \Omega} \left| \mathbb{E}_{i \sim \mathcal{I}} [\mu_i(x)] - \mathbb{E}_{i \sim \mathcal{I}} [\nu_i(x)] \right| \tag{Eq (25)}
\]

\[
\leq \frac{1}{2} \cdot \sum_{x \in \Omega} \mathbb{E}_{i \sim \mathcal{I}} \left[ |\mu_i(x) - \nu_i(x)| \right] \tag{Triangle inequality}
\]

\[
\leq \mathbb{E}_{i \sim \mathcal{I}} \left[ \frac{1}{2} \cdot \sum_{x \in \Omega} |\mu_i(x) - \nu_i(x)| \right] \tag{Linearity of expectation}
\]

\[
\leq \mathbb{E}_{i \sim \mathcal{I}} [\| \mu_i - \nu_i \|_{\text{tvd}}]. \tag{Eq (25)}
\]
Entropy and $\ell_2$-norm. We relate the entropy and $\ell_2$-norm using the following two lemmas.

**Lemma A.11.** For any random variable $A$ with support size $m > 16$,

$$\|A\|_2^2 + \mathbb{H}(A) \leq \frac{1}{m} + \log m,$$

*Proof.* Let $\text{supp}(A) = (a_1, \ldots, a_m)$ and for brevity let $\mu := \text{dist}(A)$ and $\mathcal{U}$ as the uniform distribution on $\text{supp}(A)$. Suppose first that there is an element $a_i$ with $\mu(a_i) > 1/2$. Let $\Theta$ be the indicator variable for the event $A = a_i$. In this case,

$$\|A\|_2^2 + \mathbb{H}(A, \Theta) \leq 1 + \mathbb{H}(\Theta) + \frac{1}{2} \cdot \mathbb{H}(A \mid \Theta = 0)$$

(by chain rule of entropy and since conditioned on $\Theta = 1$, $A$ has no entropy)

$$\leq 2 + \frac{1}{2} \cdot \log m < \log m.$$  

(as $\mathbb{H}(\Theta) \leq 1$, Fact A.1-(1), and since $m > 16$)

As such, in this case, the first inequality holds trivially.

Let us now consider the more interesting case when $\mu(a_i) \leq 1/2$ for all $a_i \in \text{supp}(A)$. We have,

$$\log m - \mathbb{H}(A) = \mathbb{D}(A \mid \mathcal{U})$$

(by Fact A.5)

$$\geq \frac{1}{2} \cdot \|A - \mathcal{U}\|_2^2$$

(by Pinsker’s inequality (Fact A.9) and since $\|A - \mathcal{U}\|_{\text{tvd}} = \|A - \mathcal{U}\|_1/2$)

$$\geq \|A - \mathcal{U}\|_2^2$$

(as $|\mu(a_i) - \mathcal{U}(a_i)| \leq 1/2$ for all $i$ in this case and $\|x\|_2^2 \leq \frac{1}{2} \|x\|_1^2$ whenever all $x_i \leq 1/2$)

$$= \|A\|_2^2 - \frac{1}{m}.$$  

(as $\|x - y\|_2^2 = \|x\|_2^2 - 2\|xy\|_2 + \|y\|_2^2$ for all $x, y$)

Rearranging the terms proves the lemma. \(\square\)

**Lemma A.12.** For any random variable $A$ with support size $m > 16$,

$$\mathbb{H}(A) \geq \log m - (\log e) \cdot m \cdot \|A\|_2^2 + \log e.$$

*Proof.* We use the same notation as in the previous proof. We have,

$$\mathbb{H}(A) = \log m - \mathbb{D}(A \mid \mathcal{U})$$

(by Fact A.5)

$$= \log m - \sum_i \mu(a_i) \cdot \log (m \cdot \mu(a_i))$$

(by the definition of the KL-divergence in Eq (24))

$$\geq \log m - \sum_i \mu(a_i) \cdot \log(\exp(m \cdot \mu(a_i) - 1))$$

(as $x + 1 \leq e^x$ for all $x$)

$$= \log m - \sum_i \mu(a_i) \cdot (m \cdot \mu(a_i) - 1) \cdot \log e$$

$$= \log m - (\log e) \cdot m \sum_i \mu(a_i)^2 + \log e$$

$$= \log m - (\log e) \cdot m \cdot \|A\|_2^2 + \log e,$$

concluding the proof. \(\square\)
**High entropy random permutations.** The following lemma shows that if we have a “high entropy” random permutation of $[m]$ with entropy just $\approx \varepsilon \cdot m$ below the full entropy, then the entropy of an *average* coordinate is very close to $\log m - \sqrt{\varepsilon}$ (as opposed to $(\log m!)/m - \varepsilon \leq \log m - 1$ which follows trivially from the sub-additivity of entropy).

**Lemma A.13.** Let $M : [m] \to [m]$ be a random permutation on $[m]$. If $\mathbb{H}(M) \geq \log m! - m/8$, then

$$m \log m - \sum_{j=1}^{m} \mathbb{H}(M(j)) \leq 4\sqrt{(\log m! - \mathbb{H}(M))m} + 3.$$

**Proof.** Let $1 \leq \ell \leq m/2$ be an integer parameter to be fixed later. Let $T \subset [m]$ be a uniformly random subset of size $\ell$, chosen independent of $M$. Then,

$$\mathbb{H}(M) = \mathbb{H}(M \mid T)$$

(as $T \perp M$ and by Fact A.1-(3))

$$= \mathbb{H}(M_T \mid T) + \mathbb{H}(M_{\overline{T}} \mid M_T, T)$$

($M_T$ denotes $M(i)$ for all $i \in T$, and $\overline{T}$ is the complement of $T$)

$$= \mathbb{E}_T[\mathbb{H}(M_T \mid T = T)] + \mathbb{H}(M_{\overline{T}} \mid M_T, T)$$

(by sub-additivity of entropy Fact A.1-(4))

$$\leq \mathbb{E}_T \left[ \sum_{i \in T} \mathbb{H}(M(i) \mid T = T) \right] + \mathbb{H}(M_{\overline{T}} \mid M_T, T)$$

(as conditioned on $T$, $M_{\overline{T}}$ is a permutation on $m - \ell$ elements and by Fact A.1-(1))

$$= \frac{\ell}{m} \sum_{i=1}^{m} \mathbb{H}(M(i)) + (\log(m - \ell)!).$$

(by the choice of $T$)

Therefore,

\[ \text{LHS in Lemma A.13} = m \log m - \sum_{i=1}^{m} \mathbb{H}(M(i)) \leq m \log m + \frac{m}{\ell} \cdot (\log(m - \ell)! - \mathbb{H}(M)) \]

(using the above inequality and by re-arranging the terms)

\[ = m \log m + \frac{m}{\ell} \cdot (\log m! - \mathbb{H}(M)) - \frac{m}{\ell} \cdot \log(m - 1) \cdots (m - \ell + 1) \]

\[ \leq m \log m + \frac{m}{\ell} \cdot (\log m! - \mathbb{H}(M)) - \frac{m \log(m - \ell)\ell}{\ell} \]

\[ = m \log \left(1 + \frac{\ell}{m - \ell}\right) + \frac{m}{\ell} \cdot (\log m! - \mathbb{H}(M)) \]

\[ \leq \frac{(\log e \cdot m \cdot \ell)}{m - \ell} + \frac{m}{\ell} \cdot (\log m! - \mathbb{H}(M)). \]

By setting $\ell := \left\lceil \sqrt{(\log m! - \mathbb{H}(M))m} \right\rceil \leq m/2$, we have,

\[ \text{LHS in Lemma A.13} \leq \frac{(\log e \cdot m \cdot \ell)}{m - \ell} + \frac{m}{\ell} \cdot (\log m! - \mathbb{H}(M)) \quad \text{(by the above inequality)} \]
\[ \leq \frac{(\log e) \cdot m \cdot (\sqrt{\log m! - \mathbb{H}(M)} m + 1)}{m - \frac{m}{2}} + \sqrt{\log m! - \mathbb{H}(M)m} \]
\[ \leq 4\sqrt{\log m! - \mathbb{H}(M)m} + 3. \]

**Weighted sum of entropy terms.** We also have the following technical lemma that bounds the weighted sum of entropy terms (some intuition on the statement is given right after the lemma).

**Lemma A.14.** Suppose \( M \) is a random variable with uniform distribution over \([t]\) and \( A \) is a random variable with support size at most \( \ell \) and distribution \( \alpha(A) \). Then for any arbitrary function (not necessarily a distribution) \( \beta : \text{supp}(A) \rightarrow [\epsilon, 1] \), we have:

\[ \sum_{A \in \text{supp}(A)} \beta(A)^{-1} \cdot \alpha(A)^2 \cdot \mathbb{H}(M \mid A = A) \geq \log t - \log (\ell/\epsilon). \]

For a high level intuition of the lemma, consider the case when \( \alpha = \beta \); in this case, the lemma simply says that conditioning on the random variable \( A \) cannot reduce the entropy of \( M \) from \( \log m \) more than the entropy of \( A \) which follows immediately from the chain rule (Fact A.1-(5)).

**Proof of Lemma A.14.** As \( M \) is the uniform distribution, for any choice of \( a \in \text{supp}(A) \),

\[ \mathbb{H}(M \mid A = A) = \log t - \mathbb{D}(M \mid A = A \mid M) \]

(by Fact A.5)

\[ \geq \log t - \log \frac{1}{\Pr(A = A)} \]

(by Fact A.6)

\[ = \log t + \log \alpha(A). \]

Therefore,

\[ \text{LHS of Lemma A.14} \geq \frac{\sum_{A \in \text{supp}(A)} \beta(A)^{-1} \cdot \alpha(A)^2 \cdot (\log t + \log \alpha(A))}{\sum_{A \in \text{supp}(A)} \beta(A)^{-1} \cdot \alpha(A)^2} \]

\[ = \log t + \frac{\sum_{A \in \text{supp}(A)} \beta(A)^{-1} \cdot \alpha(A)^2 \cdot \log \alpha(A)}{\sum_{A \in \text{supp}(A)} \beta(A)^{-1} \cdot \alpha(A)^2} , \]

and thus we only need to show that

\[ \sum_{A} \frac{\alpha(A)^2}{\beta(A)} \cdot \log((\ell/\epsilon) \cdot \alpha(A)) \geq 0. \]  

(26)

For each term, the derivative (w.r.t. \( \alpha(A) \)) is equal to:

\[ \frac{\alpha(A)}{\beta(A)} \cdot (2 \log((\ell/\epsilon) \cdot \alpha(A)) + 1). \]

Note that

- when \( \alpha(A) < \epsilon/\ell \), the derivative is at most \( \alpha(A)/\beta(A) < 1/\ell; \)
- when \( \alpha(A) > 1/\ell \), the derivative is at least \( \alpha(A) \cdot (2 \log(1/\epsilon) + 1) > 1/\ell. \)
If there is any \( A \in \text{supp}(A) \), where \( \alpha(A) < \epsilon/\ell \), there must also be a \( A' \) such that \( \alpha(A') > 1/\ell \) (since the average value of \( \alpha(A) \) is at least \( 1/\ell \)). Let \( \delta = \min \{ \alpha(A') - 1/\ell, \epsilon/\ell - \alpha(A) \} \).

By increasing \( \alpha(A) \) by \( \delta \) and decreasing \( \alpha(A') \) by \( \delta \), the left-hand-side of (26) could only decrease by the bounds on the derivative; moreover, \( \alpha(\cdot) \) still remains a valid distribution.

By doing such adjustments for at most \( 2\ell \) times, we obtain a new distribution \( \hat{\alpha} \) such that \( \hat{\alpha}(A) \geq \epsilon/\ell \) for all \( A \in \text{supp}(A) \), and throughout this process, we only decrease the LHS of (26). On the other hand, for the distribution \( \hat{\alpha} \), since \( \hat{\alpha}(A) \geq \epsilon/\ell \), every term in the LHS of (26) is nonnegative and thus (26) holds. This implies that it must also hold for the \( \alpha(\cdot) \) as well, proving the lemma.

\[ \boxed{\text{B Further Background on Boolean Hidden Hypermatching}} \]

\textbf{Origin of the Problem}

The Boolean Hidden Hypermatching BHH problem of Verbin and Yu [85] is inspired by the Boolean Hidden Matching problem of Gavinsky \textit{et al.} [47] (that corresponds to BHH with \( t = 2 \)), which itself is a Boolean version of the Hidden Matching problem of Bar-Yossef \textit{et al.} [12]. The original motivation behind the latter two works was proving separations between quantum versus one-way randomized communication complexity. However, it was observed in the work of Verbin and Yu that these problems are also highly applicable for proving streaming lower bounds. Thus a main conceptual contribution of [85] was the introduction of BHH for proving streaming lower bounds.

It is worth mentioning that stronger separations between quantum versus \textit{two-way} randomized communication complexity has also since been proven, for instance for the Shifted Approximate Equality (Shape) problem of Gavinsky [46]. While these problems do not suffer from the weakness of BHH which is only tailored to one-way protocols, to our knowledge, they have also not found applications for proving streaming lower bounds. This is primarily due to the fact that such problems are “much harder” than most streaming problems of interest and hence not suitable for the purpose of a reduction. Indeed, most graph streaming problems such as gap cycle counting, MAX-CUT, and property testing of connectivity or bipartiteness, tend to become “easy” with large number of passes over the input (only as a function of \( \epsilon \)); such tradeoffs provably cannot be captured by communication problems that are hard even in the two-way model (see, e.g. [7]).

\textbf{Other Variants}

There are multiple other variants of BHH studied in the literature depending on the application.

Kapralov, Khanna, and Sudan [62] introduced the Boolean Hidden Partition problem in which the input to Bob, instead of a hypermatching, is a random sparse graph with no short cycles. This problem was then used by [62] to prove an \( \Omega(\sqrt{n}) \) space lower bound for better-than-2 approximation of MAX-CUT. Inspired by this problem, Kapralov, Khanna, Sudan, and Velingker [63] introduced the Implicit Hidden Partition problem, which is a multi-party communication problem and its definition is rather different from the setup of BHH. Communication complexity of this problem was first analyzed in [63] for three players which gave an \( \Omega(n) \) space lower bound for \( (1 + \Omega(1)) \)-approximation of MAX-CUT. Subsequently, Kapralov and Krachun [64] proved a communication lower bound for this problem for arbitrary number of players which culminated in the optimal lower bound of \( \Omega(n) \) space for better-than-2 approximation of MAX-CUT.

Yet another variant of BHH was introduced by Guruswami and Tao [53] as the \( p \)-ary Hidden Matching problem in which Alice, instead of a Boolean vector \( x \in \{0, 1\}^n \), is given a vector over a larger field \( \mathbb{F}_p^n \) for some arbitrary prime \( p \) and we use computation over \( \mathbb{F}_p \) (instead of \( \mathbb{F}_2 \), i.e.,
XOR) to interpret the matching $Mx$. This problem was then used in [53] to prove lower bounds for streaming unique games and other CSP-like problems.

Finally, BHH and related problems have been studied in the multi-party simultaneous communication model in which the input is partitioned across multiple players who, simultaneously with each other, send a single message each to a central coordinator that outputs the final answer [9,57].

Streaming Lower Bounds

We now list several key implications of BHH for proving streaming lower bounds but note that this is not a comprehensive list of such results.

- **Approximating MAX-CUT**: One can approximate MAX-CUT to within a factor of 2 in $O(\log n)$ space (by counting the number of edges) and to $(1+\varepsilon)$ in $O(n/\varepsilon^2)$ space (by maintaining a cut sparsifier); see, e.g. [62]. A direct reduction from BHH can prove that any $(1 + \varepsilon)$-approximation single-pass algorithm requires $n^{1-O(\varepsilon)}$ space [62,67], and a series of work building on this approach [62–64] culminated in the optimal lower bound of $\Omega(n)$ space for less-than-2 approximation in [64].

- **Streaming unique games and CSPs**: Motivated in parts by lower bounds for MAX-CUT, other constrained satisfaction problems (CSPs) have been studied in [53, 54]. For instance, [54] proves a lower bound of $\Omega(\sqrt{n})$ space for better-than-(7/8) approximation of maximum acyclic digraph using a reduction from BHH and [53] proves an $\Omega(\sqrt{n})$ space lower bound for streaming unique games using a reduction from $p$-ary Hidden Matching problem discussed above.

- **Estimating matching size**: There is an active line of work on estimating matching size in graphs (in particular planar and low arboricity graphs) [9,35,41,61,75]. A reduction from BHH can prove an $\Omega(\sqrt{n})$ space for better-than-(3/2) approximation in planar graphs [41] which was improved to $(1+\varepsilon)$-approximation in $n^{1-O(\varepsilon)}$ space for low-arboricity graphs in [26]. These results were further extended to super-linear-in-$n$ lower bounds for dense graphs by [9].

- **Estimating matrix rank and Schatten norms**: Using an elegant connection between matching size estimation and estimating rank of $n$-by-$n$ matrices (dating back to a work of Lovász [71]), the aforementioned lower bounds for matching size estimation were also extended to algorithms that can estimate rank of $n$-by-$n$ matrix in the streaming setting [9,26]. These results were further generalized to all Schatten $p$-norms of $n$-by-$n$ matrices for all values of $p \in [0, +\infty)$ which are not even integers in [70] using another reduction from BHH (see also [22] for another variant of these results).

- **Streaming Property Testing**: Solving property testing problems in the streaming setting (as opposed to the more familiar query algorithms) has been receiving increasing attention lately [37, 55, 77, 81] starting from the pioneering work of [55]. On this front, BHH has been used to prove $n^{1-O(\varepsilon)}$-space lower bounds for $\varepsilon$-property testing of connectivity, cycle-freeness, and bipartiteness [55].

In addition, the BHH problem has also been used to prove lower bounds in other settings such as distribution testing [5], distributed computing [44], property testing [11], and sketching [57]. We leave this as an interesting open direction to explore the implications of our multi-round lower bounds for OMC in these other models.